RECONSTRUCTING OBJECTS FROM PROJECTIONS

by

WILLIAM CLEMENT KARL

S.B., Massachusetts Institute of Technology (1983)
S.M., Massachusetts Institute of Technology (1984)
E.E., Massachusetts Institute of Technology (1990)

SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1991

Signiture of Author_____
Department of Electrical Engineering and Computer Science
January 25, 1991

Certified by_____
George C. Verghese
Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by_____
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Reconstructing Objects from Projections

by

William Clement Karl

Submitted to the Department of Electrical Engineering and Computer Science
on February 4, 1991, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis addresses reconstruction problems arising in the determination of an object from a set of noisy silhouette projections. Examined in detail are the orthogonal projection and reconstruction of smooth, convex surfaces from curvature information, and the mathematically related problem of estimating ellipsoids from projections. These two problems share a common underlying algebraic structure in their involvement of a positive semi-definite (PSD) symmetric matrix. A linear mapping between symmetric matrices is defined which captures the projection structure. The nature of the PSD constraint is illuminated and exploited to develop PSD-constrained matrix estimation algorithms. Extensions to symmetric matrices with arbitrary interval matrix bounds of the form $\underline{X} \leq X \leq \overline{X}$ are provided. The resulting matrix estimates provide a complete solution to the original ellipsoid and curvature reconstruction problems. The inclusion of a dynamic relationship is straightforward, and is demonstrated for the case of ellipsoids. The techniques developed are applicable to other linear symmetric matrix problems.

Consistency and smoothness of reconstructions based on discrete support hyperplane measurements are also investigated. Such support measurements arise from 1-dimensional projections of objects. The classical theory of surfaces and support functions is extended to the discrete, general dimensional case, enabling the development of a linear, local consistency test for a set of support measurements. The close tie between consistency and curvature is exploited to develop various discrete definitions of surface curvature for use as measures of smoothness. Global definitions of smoothness based on the isoperimetric inequality are also provided.

This thesis contributes to the field of reconstruction from projections by extending and clarifying previous work on the planar case, by providing computationally attractive solutions to certain problems, and by suggesting new approaches to existing issues.

Thesis Supervisor: George C. Verghese
Title: Associate Professor of Electrical Engineering

Acknowledgments

There are many people who have contributed to the success of this thesis. I cannot credit George Verghese, my thesis advisor, enough for his insights, suggestions, and guidance. George has indelibly shaped my outlook, focus, and approach to this and all research. His high academic standards have served as a guide for my own. I would also like to thank Profs. Alan Willsky and Berthold Horn for their helpful comments and input at various stages in this research.

I would also like to thank my friends and associates in LEES, who have helped me both directly and indirectly. My contemporaries, Aleks Stankovic, Miguel Velez, and Wayne Hagman have shared both their time and evenings with me. Larry Jones, my officemate, actually seemed to understand my sense of humor and certainly put up with my mess. Vivian Mizuno was always ready to lend me a helping hand, as she did during the last day of writing. Jeffrey Lang has always been willing to help me sort out my quandaries. It has been a distinct pleasure and privilege to spend time with the faculty, students, and staff of LEES and I will miss them all.

I could not have endured the many trials and tribulations of this endeavor without the ongoing support of my friends and extended family. They have kept me sane through the years: Mike and Janine Nell (who kept me from starving), Tom Lee (my cohort in comedy), Bob Harris (my oldest friend), Andy Cohen (who met "the new kid"), Dan McCarthy (my brother), Lisa Swanson and Barri Clark (who showed me what a picnic was), Elizabeth Anderson (who was always up for the Border Cafe), Joe Pennario (who always had a laugh for me), Lisa Sievers (with whom I shared many a lunch), and many others. Peyman Milanfar has been both a great friend and a great aid, helping me to sort out numerous technical points and distracting me from my typing.

I wish to acknowledge the various agencies and companies that have supported both me and this work. This research has been partially supported by the National Science Foundation under Grant ECS-8700903. I have also been the fortunate recipient of generous fellowships from the Kodak Company, the Center for Intelligent Control Systems under U.S Army Research Office Grant DAAL03-86-K-0171, and the Department of Electrical Engineering and Computer Science.

Thank you one and all.

to Carolyn, Mom, and the memory of my father.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

This thesis addresses certain reconstruction problems arising in the determination of an object from a set of noisy shadow projections. Shadow projections appear in such disciplines as geophysics, medicine, oceanography, robotics, and computer graphics, arising when information about only the *boundary* or *support* of a projected object is used as an observation. In addition to problems that are directly shadow based, many practical situations are modeled as such out of convenience or necessity. This is particularly true in situations of high noise or limited information, resulting in projection observations which are effectively shadows. Finally, for many fundamentally non-shadow problems, a preliminary step of object boundary estimation using shadow information can greatly aid the achievement of the final reconstruction goal.

In this thesis we are primarily interested in the relationship between the algebraic and geometric structure of certain problems arising from the shadow projection of objects. We expose and exploit such underlying algebraic structure to: 1) unify and extend existing reconstruction results; 2) provide concise solution characterizations for these problems; 3) develop algebraic constraints reflecting fundamental geometric object properties; and 4) suggest natural, geometrically constrained reconstruction algorithms.

Three related problems are treated in detail. The first of these involves the orthogonal projection and reconstruction of smooth, convex hypersurfaces from curvature information. This problem mainly centers around the local properties of a surface and its projections, but also relates to global issues of overall shape. The second problem we examine arises from the mathematically related task of estimating ellipsoids

from their projections. Ellipsoids are shapes commonly used in many applications to capture basic properties of object geometry, such as orientation and eccentricity. This problem and the previous one share a common underlying algebraic structure in that both involve a defining positive semi-definite (PSD) symmetric matrix, resulting in our examination of these *algebraic* objects. Our third problem concerns the consistency and smoothness of reconstructions based on discrete support measurements. One-dimensional shadow projections of objects yield such support measurements. In particular, we examine the reconstruction of ellipsoids from such observations. Throughout this work we concentrate on object *boundary* reconstruction, as distinct from those disciplines attempting to discern information (such as density variations) about the interior of an object.

This chapter is organized as follows. In the next section we discuss shadow projections and their relation to this work. The specific contributions of this thesis are discussed in Section 1.3. Section 1.4 gives a summary of the organization of the thesis.

1.2 Shadow Projections and Object Support

The relationship between objects and their shadows or orthogonal projections is of interest in a variety of disciplines and applications, including computer graphics, low dose tomography, and robotics. Some applications are concerned with the forward problem of determining the shadow of an object on a surface, given the shape of the object. This is the goal, for example, in image synthesis for computer graphics [1, 2] and automatic drawing generation for a part [3, 4]. Other applications are concerned with the inverse problem of recovering the boundary shape of an object from a series of its projections. This recovery can be direct or indirect. The projections can be used to directly reconstruct a shape approximating the underlying generating one, as is done in medical applications, or we can use our observations in a recognition framework to choose one from among a set of candidate objects, thus indirectly recovering the shape, as is done for target recognition tasks [5, 6, 7]. Our primary interest is in the direct reconstruction framework, though our results may be used for the recognition problem as well.

The generation of the shadow observations considered here comes about in different ways. Aside from directly obtained shadow images, projection data often contains little more information than the outline of the object of interest [7]. This is true in certain instances of non-destructive testing, in low-dose transmission X-ray tomographic

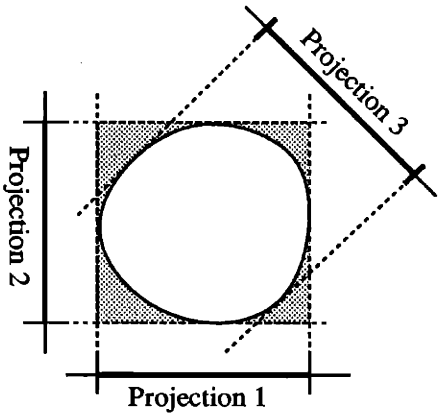

Figure 1-1: Illustration of a projection measurement.

images [8, 9], in certain robotic applications involving spatial probing or grasping, and in noisy range images [10, 11, 6, 5], to name a few. In all these cases the structure of the observed data is effectively reduced to a projected outline. Even when this is not the case, some applications attempting to reconstruct the interior of an object benefit from a preliminary step of boundary extraction and reconstruction using only such outline information. This approach has proven particularly helpful in reflection tomography from laser range data [12] and has also been used in the case of low dose tomography [8]. Mathematically, these shadow observations can be obtained through a number of projection geometries, including orthogonal projection, perspective projection, and spherical projection [13, 14, 15]. Unless otherwise stated, in this thesis we will mean an orthogonal shadow projection, as illustrated in Figure 1-1, when we refer to a projection.

Once such a set of observations is obtained, it can be used in different ways to constrain object shape. Sometimes observations are used in a simple and straightforward manner [16, 17], where each shadow is used to find a bounding volume for the object and the combination of all such bounds forms an approximation to the object. Often, however, the projections are used in conjunction with a model of the object [9, 8, 18] or a representation in some transformed space [15, 19, 20], such as the Gaussian sphere. These models and representations are chosen for their convenience when working with projection information, and because they provide simple relationships between the object in question and its projections. It is this latter approach that we take, using projection observations together with certain object models and representations.

One such representational tool that is used is a map of an object to the Gaussian sphere [21]. The Gaussian sphere is a unit sphere where each point on the sphere corresponds to points of an object with the same surface normal orientation. Thus a polyhedron is represented by a finite set of points on the Gaussian sphere. This map to the Gaussian sphere is often combined with a map of some corresponding object property, such as distance to the face for the case of polyhedra. In [7] this additional mapping was done using local curvature information. For convex surfaces, it is a classical result of differential geometry [22, 23, 24] that this curvature information for all normal orientations uniquely determines the surface shape to within translation. Such a representation thus uses the local curvature information to capture global surface properties.

Another approach to reconstruction from projections is provided by methods that make direct use of a geometric object model. The object is parameterized directly by a relatively small number of embedded unknown parameters. The work in [9, 25] on tomographic density reconstruction under conditions of low signal-to-noise ratio is representative of this approach. This work (for the planar case) used an elliptical model to capture object center, radius, eccentricity, and orientation. For many problems, such general object information is all that is required and the generation of a detailed image, as is generally done for conventional medical tomography, is not necessary. The advantage of this approach for the tomographic problem is that one can expect to do better at estimating only a few parameters then at estimating the 65,000 pixels that typically exist in an image. The approach also has demonstrated itself to be extremely robust in the presence of modeling errors [9].

The geometric modeling approach in [9, 25] has been expanded upon in the works of Bresler and Macovski [26, 18] and in the work of Prince [8]. The first of these used collections of cylinders to model blood vessels in digital subtraction angiography. These cylinders each have an elliptical cross section and their spatial evolution is modeled by a dynamic equation in the parameters of the planar ellipses. In projection, each cross section resulted in a 1-dimensional observation of both density and extent. The work in [8] extended the geometric modeling formalism through the incorporation of prior knowledge. In particular, one part of the work used an elliptical model to capture prior shape information regarding object eccentricity and orientation. Another part was concerned with the consistency and smoothness of the reconstructed boundary. This work was confined to planar problems.

For convex objects, the ones of primary interest here, projection observations are

equivalent to information about object extent in particular directions. This distance to an object's tangent hyperplane of a given direction is termed its *support value* in that direction. Each such observation tells us something about where the object can and cannot be. Consider the example in Figure 1-1. Given just projections 1 and 2 we know that the object must be contained in the shaded region, the projections defining the extent of the object in particular directions. Clearly projection 3 cannot be arbitrary if it is to be consistent with the other two. For example it cannot be outside the shaded region. Thus the three projections and the support values that they imply must satisfy a consistency constraint. Such questions for a continuous set of measurements have a classical origin [27]. In [8] the corresponding problem for the *discrete* planar case was examined, and a linear constraint provided for consistency of support samples at equal angles. One goal of the treatment in [8] was to reconstruct an optimal consistent set of measurements for use in other reconstruction tasks. This work was given a distributional interpretation in [12]. Again, the work in [8, 12] is limited to the planar case.

1.3 Contributions

This thesis makes contributions in several areas. One is our unification and extension of existing planar or low dimensional results to a general setting. Often these results had existed only for the planar case and our extensions to the 3-dimensional case are immediately useful. The curvature results of Chapter 4 and support consistency results of Chapter 6 are examples of such results. Further, existing shadow based projection approaches have, at best, confined themselves to objects in three dimensions and shadows in two dimensions. We can imagine, however, situations where such restrictions might be restrictive and a more general formulation could be required. Any reconstruction problem involving a 3-dimensional object that evolves in time, such as imaging of the beating heart [28, pp. 275], is inherently 4-dimensional. Indeed, much work related to the Radon transform is already done in such a general dimensional setting [29, 30]. The general settings in which we treat our problems serve both to directly address this weakness of earlier work and to provide a unification of the present planar work in the area.

A second area of contribution is given by our solution characterizations. We provide concise solutions to the problems of curvature and ellipsoid reconstruction from projections. The existence of these solutions depends on the rank properties of

a certain matrix and is thus simple to check. The corresponding solutions to these problems are related to linear matrix equations. Our linear formalism allows simple computation of the solutions to a given problem. These linear models are also suitable for recursive implementation and constrained reconstruction.

A third area of contribution is to the problem of constrained shape reconstructions. Our development in Chapter 3 provides novel, geometrically based algorithms for constrained matrix reconstruction under arbitrary matrix interval bounds, including eigenvalue bounds. These matrix algorithms and their constraints may in turn be used to reflect the geometric properties of objects in the corresponding surface reconstruction problems. In particular, the prescription of a range of allowed boundaries for the ellipsoid reconstruction problem and the prescription of a family of allowed surface shapes for the curvature reconstruction problem fall into this category. Also, in Chapter 6 a local linear consistency constraint for a set of support measurements is provided for the 3- and higher-dimensional case. Such a linear constraint is easily incorporated into existing support based reconstruction algorithms. Finally, our development of simple measures of discrete curvature and smoothness in the same chapter allows the incorporation of smoothness constraints in support based reconstructions of 3-dimensional objects.

1.4 Organization

This thesis is organized as follows. In Chapter 2 we introduce a particular class of linear projection mappings between symmetric matrices, together with an isometric isomorphism of this problem to a convenient vector space setting. The definition of this mapping is motivated by the problems of curvature and ellipsoid reconstruction that we investigate. Following this treatment of matrix projections, in Chapter 3 we examine the inverse problem of reconstructing a symmetric matrix from one or more of its projection mappings. The material in these two chapters is common to much of the subsequent work. Chapter 4 examines the relationship between the surfaces of smooth, strictly convex objects and the surfaces of their orthogonal projections, or shadows. This problem is algebraically related to the matrix issues of the first two chapters. In Chapter 5 we treat the problem of projecting and reconstructing an ellipsoid. A common underlying algebraic structure between this problem and the problem of curvature based reconstruction relates these issues to the problem of symmetric matrix estimation treated earlier in Chapter 2. In particular, we examine

the case of 1-dimensional projections of ellipsoids, yielding support sample observations. Chapter 6 develops these notions more fully by examining the consistency and smoothness of reconstructions based on a discrete set of support samples. In Chapter 7 we suggest promising directions and problems for further work. Chapter 8 summarizes our results.

Chapter 2

Symmetric Matrices: Properties and Projections

2.1 Introduction

In this chapter we introduce a particular class of linear projection mappings between symmetric matrices. The definition of this mapping is motivated by the applications in which we are interested, and thus comes from physical considerations, as we describe below. The material to be presented in this chapter is common to much of the subsequent development and knowledge of these results will serve us well.

The problems concerning symmetric matrices treated in this chapter appear in many different guises in physical problems. For example, we show in Chapter 4 that the relationship between the curvature of a convex surface and the curvature of the boundary of its orthogonal projection is determined by the behavior of a certain symmetric matrix under a corresponding symmetric projection mapping. In Chapter 5 we show how the problem of projecting and estimating (static and dynamic) ellipsoids may also be treated as one of these problems in symmetric matrix projection and estimation. Another problem fitting this framework is the determination of the (symmetric) conductivity matrix of a network from voltage probes [31]. At the root of all these problems lies an underlying, desired symmetric matrix, of which we have partial information through certain projection observations. We mention these physical ties as they evolve here, but save deeper treatment for the appropriate chapter.

We start by defining a notion of symmetric projection mapping for symmetric matrices that is central to our work. The definition of this mapping is motivated by

the applications which were mentioned above, and thus derives from physical considerations. Different formalisms for relating a matrix and its projection are presented and critiqued, resulting in our choice of a particular isometric isomorphism as begin most useful.

For many of the problems involving a symmetric matrix to be physically meaningful, the underlying matrices are required to be positive semi-definite. The ellipse and curvature problems mentioned above are two of these. As a consequence, we next examine the nature of this constraint on the set of symmetric matrices in the context of the different formalisms above. The set of PSD matrices forms a convex cone, but one that is not finitely generated. Aspects of the geometry of this cone are examined, and in particular, finite approximations to it are developed yielding *linear* constraints. Such constriants are useful for estimation algorithms, are are treated in the Chapter 3.

Finally, we note that the methods and formalisms developed in this chapter are actually applicable to a much wider class of problems than just the estimation of a symmetric matrix from a series of our specially defined symmetric projection mappings. The formalism we develop to handle our particular linear symmetric matrix problem is actually a convenient way to solve many linear problems involving a matrix argument. Further, the insight we develop into the nature of the positive semi-definite constraint may be extended to arbitrary eigenvalue bounds. In combination with the results of Chapter 3 we thus have a way of solving many matrix problems that are linear in their matrix argument. As our primary interest is in our particular symmetric case, we pursue that line of development and point out the necessary extensions to this wider class of problems along the way.

2.2 Projection Mapping

In this section our notion of the projection mapping of a symmetric matrix is made precise and its use motivated. Two examples suffice here to create the tie both between our applications and their underlying symmetric matrices and with the particular notion of projection that we use. More detail on the specific applications is presented in later chapters.

To start, consider the relationship between the curvature of a convex surface and that of the boundary of its projection, alluded to above. This situation is sketched in Figure 2-1. We may take the curvature of a surface at a point to be the symmetric

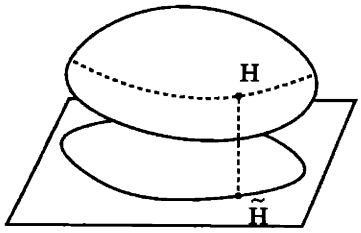

Figure 2-1: Curvature projection

Hessian matrix H of a local height function at the point. The Hessian is the matrix of second partial derivatives of a function. If the Hessian matrix of the orthogonal projection of the surface at a point is given by $\tilde{\mathsf{H}}$, then we show in Chapter 4 that the relationship between this projected curvature and the curvature of the surface at the point's preimage is given by the equation:

$$\tilde{\mathsf{H}}^{-1} = S^T \mathsf{H}^{-1} S \tag{2.1}$$

for some S which is determined by the projection. Thus under orthogonal projection, the symmetric matrices $\tilde{\mathsf{H}}^{-1}$ and H^{-1} are related by the quadratic form (2.1).

In another vein, consider the orthogonal projection of an ellipsoid. We may represent the ellipsoid by a symmetric matrix, since for any symmetric positive semi-definite matrix E, the set $\{x | x^T u \leq \sqrt{u^T \mathsf{E} u},\ \forall \|u\| = 1\}$ defines an ellipsoid. We may thus use E to *define* the ellipsoid. Projecting this ellipsoid orthogonally onto a subspace yields another lower dimensional ellipsoid, which we may similarly represent within the subspace by $\tilde{\mathsf{E}}$. The original ellipsoid and its projection are again related by the equation:

$$\tilde{\mathsf{E}} = S^T \mathsf{E} S \tag{2.2}$$

where again the matrix S is a function of the projection subspace only.

The above examples should suffice to give a feel for how our physical problems are related to the theoretical issues of this chapter. The forms of the above equations motivate the following definition of projection mapping for symmetric matrices:

Definition 1 (Symmetric Projection Mapping) *Given a symmetric $n \times n$ matrix X and a general $n \times m$ matrix A ($m \leq n$) we define the symmetric projection mapping of X by A to be the matrix:*

$$Y = A^T X A \tag{2.3}$$

Thus Y itself is a $(m \times m)$ symmetric matrix. For convenience, we often refer to Y as simply a projection of X. Note that this mapping represents a true projection only if the columns of A are orthonormal. We are mainly interested in the case when $m < n$ and the matrix Y is *smaller* than X, the mapping representing a loss of information. Most of our results are easily extended to the case where $n < m$.

Note that the entries of Y are linear functions of the entries of X, thus the operation (2.3) represents a *linear transformation* of the entries of X. As such, a matrix representation of this operator must exist. Such equivalent representations of the operation in (2.3) will be very useful in the sequel and are examined next.

2.2.1 Equivalent Representations

Kronecker Approach

One way that has been used to represent the operator (2.3) is through the use of the Kronecker and vec operators [32, 33, 34, 35]. Given two matrices, A and B, the Kronecker product of A and B (notation $A \otimes B$) is defined to be the matrix formed by all products of entries of A with B, so that $A \otimes B = [a_{ij}B]$. For a matrix A, the vec of A, notation $\mathsf{vec}(A)$ is the vector formed by stacking the columns of A one above the other with the first column at the top and the last at the bottom. With this notation it is straightforward to show that we may rewrite (2.3) as:

$$\mathsf{vec}(Y) = (A^T \otimes A^T)\,\mathsf{vec}(X) \tag{2.4}$$

where $(A^T \otimes A^T)$ is an $m^2 \times n^2$ matrix, $\mathsf{vec}(Y)$ is a m^2 vector, and $\mathsf{vec}(X)$ is an n^2 vector of the elements of X (recall A is $m \times n$). This form of (2.3) confirms the linear nature of the relationship. Unfortunately, while the vector $\mathsf{vec}(X)$ has n^2 entries, the corresponding matrix X, being symmetric, has only $n(n+1)/2$ independent elements. Not all the elements of $\mathsf{vec}(X)$ are independent as there is hidden structure in the vector $\mathsf{vec}(X)$. The same thing can be said of both $\mathsf{vec}(Y)$ and $(A^T \otimes A^T)$, making work with this form difficult.

One way around the above difficulties is to use the symmetry condition to reduce the dimension of the vectors on the left and right hand sides of (2.4) from the original m^2 and n^2 elements to only the independent $m(m+1)/2$ and $n(n+1)/2$ elements, respectively. This reduction is done by combining certain redundant rows and columns of the matrix $(A^T \otimes A^T)$ and eliminating certain entries of $\mathsf{vec}(X)$ and $\mathsf{vec}(Y)$. This

approach has been taken by Magnus et. al [33] and others [32, 34], who define an $n(n+1)/2 \times n^2$ *elimination* or deletion matrix L so that $L\mathsf{vec}(X)$ is sans the supradiagonal elements of X (which have been eliminated by the action of L). For example, for $n = 3$ L has following form:

$$L = \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & & & & & & \\ 0 & 1 & 0 & & 0 & & & 0 & \\ 0 & 0 & 1 & & & & & & \\ \hline & & & 0 & 1 & 0 & & & \\ & 0 & & 0 & 0 & 1 & & 0 & \\ \hline & & & & & & 0 & 0 & 1 \end{array}\right] \tag{2.5}$$

Other operators of this type are defined, such as an inverse operator for L, termed the dilation matrix D, essentially a certain left inverse of L. Using these auxiliary matrices, the equation (2.4) may be compressed to contain only the essential elements of X and Y. The relationship between X and Y may then be investigated, using the resulting matrices and the properties of the compression and dilation operators. In fact, the authors of [33] give an extensive treatment of pertinent properties of these operators and their applications with this approach in mind. Unfortunately, even this treatment is far from transparent, and the relationship of the resulting matrices are hard to discern [36]. This observation leads us to develop other ways of coping with (2.3).

2.2.2 The Symmetric Vector Space S

We overcome the above difficulties by embedding the problem in a natural way in the vector space of symmetric matrices. In this space, each symmetric matrix is a vector and the mapping (2.3) has a matrix representation. Unlike (2.4), however, this matrix does not have structural redundancies because the entries of the vectors now each represent an independent direction in the space.

The set of $n \times n$ symmetric matrices together with the inner product $\langle A, B\rangle_n \equiv \mathrm{tr}(A^T B)$ defines an $n(n+1)/2$-dimensional Euclidean vector space which we denote by S_n (note that this inner product induces the Frobenius norm on a matrix $\langle A, A\rangle^{1/2} = \|A\|_F$). Suppose the set of symmetric matrices $\{ M_\ell^{(n)} \,|\, 1 \leq \ell \leq n(n+1)/2, \langle M_i^{(n)}, M_j^{(n)}\rangle_n = \delta_{ij}\} = \{M_\ell^{(n)}\}$ is an orthonormal basis for this space. Each $n \times n$ symmetric matrix can be represented with respect to this basis by a vector

in $R^{n(n+1)/2}$ and conversely, each vector in $R^{n(n+1)/2}$ specifies a matrix obtained as the weighted sum of the basis elements $\{M_\ell^n\}$ where the weights are contained in the vector. A suitable standard basis is given in Appendix 2-A along with the 2×2 case as an example. We represent this mapping from the symmetric subset of $R^{n \times n}$ to $R^{n(n+1)/2}$ by the notation $x = \Xi_{\{M_\ell^n\}}(X)$. For notational convenience, we henceforth suppress the dependence of $\Xi(\cdot)$ on the particular chosen basis $\{M_\ell^n\}$. Now if $a = \Xi(A)$ and $b = \Xi(B)$, then $\langle A, B \rangle_n = a^T b$. Since the representation of each symmetric matrix given by $x = \Xi(X)$ is also unique we have the following result:

Result 1 (Isometric Isomorphism) *The mapping between a symmetric matrix X and the vector of its representation x given by $\Xi(\cdot)$ is an isometric isomorphism between the set of symmetric $n \times n$ matrices and $R^{n(n+1)/2}$.*

If x is the vector in S_n corresponding to X and y is the vector in S_m corresponding to Y, then we may represent the projection mapping between X and Y given in (2.3) by the following linear equation:

$$y = \tilde{A}x \tag{2.6}$$

where the elements of the matrix $\tilde{A}$ are given by $(\tilde{A})_{ij} = \langle M_i^{(m)}, A^T M_j^{(n)} A \rangle_m$. Note that y now has $m(m+1)/2$ elements and x has $n(n+1)/2$, as desired. Since we use the relationship between A and $\tilde{A}$ repeatedly we define it formally:

Definition 2 (The mapping Γ) *Given the $n \times m$ matrix A, and the orthonormal symmetric bases $\{M_\ell^{(n)}\}$ and $\{M_\ell^{(m)}\}$ we define the mapping Γ from A to the $m(m+1)/2 \times n(n+1)/2$ matrix $\tilde{A}$, where $(\tilde{A})_{ij} = \langle M_i^{(m)}, A^T M_j^{(n)} A \rangle_m$. We denote this mapping by the notation $\tilde{A} = \Gamma(A)$ or $A \overset{\Gamma}{\longmapsto} \tilde{A}$. Thus Γ is a map from $R^{n \times m}$* **into** *$R^{[m(m+1)/2] \times [n(n+1)/2]}$.*

Thus $\tilde{A}$ is a matrix representation of the projection mapping $A^T(\cdot)A$ with respect to the symmetric bases $\{M_\ell^{(n)}\}$ and $\{M_\ell^{(m)}\}$.

The formulation of Equation 2.6 is straightforward to use and the relationship between the variables is more transparent. For example, $\|Y - A^T X A\|_F = \|y - \tilde{A}x\|_2$, thus manipulations and minimizations done in S_n have direct connections to quantities in the original space. Some useful properties of Γ are presented in Appendix 2-B. We note here that many properties of A carry over to $\tilde{A}$ under Γ, e.g. if A is a projector (so $A = A^T$ and $AA = A$) then so is $\tilde{A}$. Such properties will be of use later.

Extensions

Conceptually, the vector space representation provided by (2.6) may be developed for any *linear* function of a matrix argument. In general, if a linear matrix operator is given by $Y = \mathcal{L}(X)$, then we may always represent this operator in the form of (2.6), where the entries of $\tilde{A}$ are now given by $(\tilde{A})_{ij} = \langle M_i^{(m)}, \mathcal{L}(M_j^{(n)}) \rangle_m$, and $\{M_j^{(n)}\}$, $\{M_i^{(m)}\}$ are any orthonormal bases for the domain and range, respectively. There always exists a matrix representation of a linear operator and the mapping to $\tilde{A}$ as defined here is one general way to find it. The definition of Γ then is no more than the definition of the linear operator (2.3) with respect to given (arbitrary) sets of bases for the symmetric spaces. Since we may phrase many of our results in terms of the matrix representation $\tilde{A}$, they are applicable to the entire class of linear matrix problems (e.g. those of the form $Y = A^T X B$, $Y = A^T X + X B$, etc. with general A, B, X, and Y) as well as for (2.3). Later we consider the solution of (2.6) with certain eigenvalue constraints on the symmetric matrix represented by x. From the above arguments, these algorithms and results are also directly applicable to the larger class of linear problems involving a symmetric X. More limited extensions of these constrained problems also appear possible to the case of non-symmetric X. As our primary interest is in the symmetric case given by (2.3), we focus on this class of problems.

Note that the Kronecker formulation (2.4) gives another matrix representation of the operator. This operator is not induced by an arbitrary choice of bases however, but rather implies its own basis. In fact for our inner product, the corresponding basis on X implied by (2.4) is comprised of matrices which are all zeros except for a single 1 in some entry. These basis elements, while orthonormal, do not form an orthogonal decomposition of the space of matrices into symmetric and skew-symmetric parts.

2.3 Positive Semi-Definite Matrices

All of the above discussion concerned the projection mapping of *general* symmetric matrices. Many physical problems fitting into the symmetric matrix projection and reconstruction framework however require on physical grounds that the underlying matrix be positive semi-definite. Positive semi-definite symmetric matrices and their properties are important in a wide variety of fields such as control theory, physics, and network theory, where they appear in such connections as Lyapunov equations, stability analyses, and covariance estimates. These reasons lead us to study the set of

positive semi-definite symmetric matrices. To this end, we first study the geometry of the PSD set and then examine some useful approximations to this PSD set. In Section 3.3 we use these insights to solve least-squares problems subject to a PSD constraint on the solution.

A symmetric matrix X is said to be positive semi-definite and denoted $X \geq 0$ if all its (necessarily real) eigenvalues are non-negative. Similar definitions hold for positive definite, negative definite and negative semi-definite, but we concentrate on the positive semi-definite case. We denote the set of $n \times n$ positive semi-definite matrices by PSD_n and the subset of $n \times n$ positive semi-definite matrices of rank r by $\mathrm{PSD}_n^{(r)}$. In the vector space of symmetric matrices S in which we have been working, the matrix X is represented by a vector. The entries of this vector are just linear combinations of the entries of the original matrix X. It is therefore more convenient to use a condition that is not phrased in terms of the spectrum of the matrix but directly in terms of the matrix elements themselves. Such well known equivalent conditions are given in the following (see e.g. [37, 38]).

Result 2 (PSD conditions) *The following statements concerning $X = X^T$ are equivalent:*

1. $X \geq 0$.
2. $u^T X u \geq 0, \ \forall \ u^T u = 1$.
3. *The principal minors of X are non-negative.*

Of special interest to us is the cases when X is 2×2 and 3×3. These cases arise in the modeling of 2- and 3-dimensional objects respectively. Applying the conditions above to these cases yields the following constraints on the entries of a matrix for it to be positive semi-definite:

Corollary 1 *If $X = X^T = [X_{ij}]$ is 2×2 then X is positive semi-definite if and only if*

$$\begin{aligned} X_{11} &\geq 0 \\ X_{22} &\geq 0 \\ X_{11}X_{22} - X_{12}^2 &\geq 0 \end{aligned}$$

If X is 3×3 then X is positive semi-definite if and only if the above conditions hold

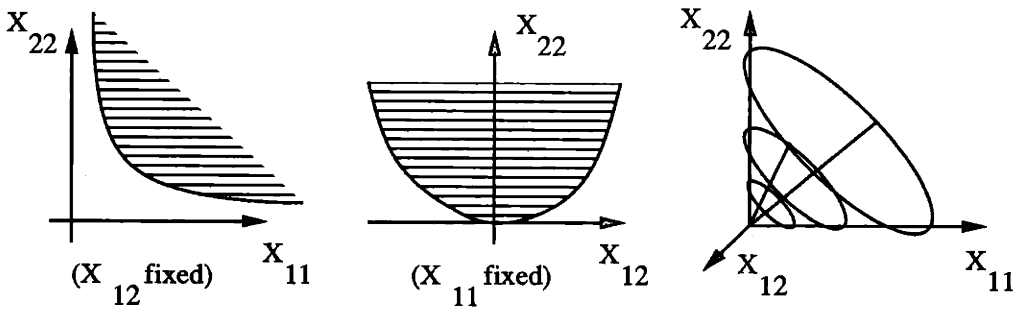

Figure 2-2: Set of PSD 2×2 matrices X

and

$$\begin{aligned} X_{33} &\geq 0 \\ X_{11}X_{33} - X_{13}^2 &\geq 0 \\ X_{22}X_{33} - X_{23}^2 &\geq 0 \\ X_{11}X_{22}X_{33} + 2X_{12}X_{13}X_{23} - X_{11}X_{23}^2 - X_{22}X_{13}^2 - X_{33}X_{12}^2 &\geq 0 \end{aligned}$$

Now, since the set of symmetric matrices is isomorphic to $R^{n(n+1)/2}$ through $\Xi(\cdot)$, we may use (2) of Result 2 to obtain the following equivalent characterization of the PSD set directly in the space S:

Lemma 1 (PSD Isometry) *The PSD set is isomorphic to the following subset of* $R^{n(n+1)/2}$*:*

$$\left\{x | x^T\mu \geq 0,\ \forall \mu \in \Xi(PSD_n^{(1)})\right\} \tag{2.7}$$

where $PSD_n^{(1)}$ *is the set of rank-1 PSD matrices and* $\Xi(\cdot)$ *is the mapping from a symmetric matrix to the vector of coefficients of its representation.*

The lemma follows from the identity $u^TXu = \langle X, uu^T\rangle = x^T\mu$, where $\mu = \Xi(uu^T)$.

For the 2×2 case we give a sketch of the set of positive semi-definite matrices (PSD_2) in the parameter space of the matrix entries (X_{11}, X_{12}, X_{22}) in Figure 2-2 to display the set involved. The set is obviously convex and cone shaped. Recall that the relationship between the entries (X_{11}, X_{12}, X_{22}) and the elements of the vector $x = \Xi(X)$ is linear, so a similar image of the PSD set holds in the space S. In fact, for our standard basis given in Appendix 2-A $x = [X_{11}\ \sqrt{2}X_{12}\ X_{22}]^T$. For the 3×3

case the set PSD_3 already resides in a 6-dimensional space and is nearly impossible to visualize! These sets turn out to have interesting structure, however, particularly in the vector space S.

We now examine the general structure of the PSD set. Insights and intuitions into the properties of the set are given together with useful characterizations. These insights and characterizations are then used to develop (polyhedral) approximations to the PSD set. These approximations in turn are used in the sequel for the solution of constrained optimization problems. We start with the general properties of the PSD set next.

2.3.1 The Set of PSD Matrices: General Properties

The general geometric properties of the set PSD_n of $n \times n$ positive semi-definite matrices is reviewed here. We start with some general observations, then apply the general theory of cones and convex sets to the case at hand to obtain more detailed characterizations of the set.

First note that the PSD set is convex, since the sum of any two PSD matrices is again PSD. It is also a closed set [39]. Using the principal minor condition above, we see that the overall order of the polynomial representing the boundary in the space of the entries (and thus in the space S) is the same as the size of the underlying matrix, n, and that this polynomial is homogeneous. Further, since the principal minors involve the determinants of symmetric submatrices of the elements, the conditions can be seen to always be linear in each of the diagonal elements X_{ii} and quadratic in the off-diagonal elements X_{ij}, $i < j$. The linearity in the diagonal matrix elements means that the bounds on these elements are one-sided: as we increase X_{ii} from $-\infty$ to $+\infty$ with the other elements fixed we only cross the positive semi-definite boundary once. It is also true that if a matrix is already positive semi-definite, then increasing its diagonal elements does not destroy this property. Thus, as can be verifed in the 2×2 example, the PSD set is open in these directions. Since the conditions for positive semi-definiteness are quadratic in each of the off-diagonal elements, as these elements are taken from $-\infty$ to $+\infty$ (with the other elements fixed), the positive definite boundary is crossed *twice* in general. Indeed, since the set is convex, this is the maximum number times we may cross the boundary through such monotonic perturbations. This behavior can be seen in the sketch for the 2×2 case above. We use these observations later in approximating the set of positive semi-definite matrices.

$\mathbf{PSD}_n$ is a Cone

We now review the properties of the PSD set as a cone. We use results and definitions from the theory of cones [39, 40, 41, 42]. A subset K of a finite dimensional vector space is said to be a convex *cone* if and only if $x, y \in K$, $\alpha, \beta \geq 0$ implies that $\alpha x + \beta y \in K$, where α and β are scalars. We can see that the set of positive semi-definite matrices forms a convex cone, since the non-negative sum of any two PSD matrices is again a PSD matrix. A cone is said to be *pointed* if $K \cap (-K) = \{0\}$. Now if X is a PSD matrix then $-X$ will be positive semi-definite if and only if X is the zero matrix, so the PSD cone is pointed. A cone is termed *full* if the interior of K is nonempty. Since the interior of the PSD cone is comprised of the (non-empty) set of positive definite matrices, the PSD cone is full. For a cone K, if $\mathsf{span}(K)$ equals the underlying space then K is termed *reproducing*. Since the span of PSD matrices is the set of symmetric matrices, PSD_n is reproducing. The *dual cone* K^* of K is defined to be $K^* = \{x | \langle x, y \rangle \geq 0, \forall\, y \in K\}$. A cone is termed *self dual* when $K = K^*$. It can be shown [43] that the real PSD cone is self-dual (if we consider the set of hermitian matrices over the complex field this is not true however). Finally a cone is termed *polyhedral* if it can be represented as the set of all non-negative linear combinations of a *finite* set of elements. Since no such finite set of generators exists for the set of positive semi-definite matrices, the PSD cone is *not* polyhedral. This is unfortunate because of its implications for the solution of constrained optimization problems. We say more about this later.

Symmetries of PSD Cone

The central direction of the PSD cone is the identity matrix and there exist certain symmetries around it. Following [44], the inner product $\langle A, B \rangle_n$ allows us to define the cosine of the angle between two matrices by

$$\cos(A, B) = \frac{\langle A, B \rangle_n}{\|A\|_F \|B\|_F}.$$

A matrix's position in the cone can be thought of as depending on its angle with respect to the central identity direction. We have the following result:

Result 3 (Tarazaga [44]) *Given a symmetric positive semi-definite matrix X:*

$$\cos(X, I) = \frac{\mathsf{Tr}(X)}{\|X\|_F \sqrt{n}}$$

$$\leq \sqrt{\frac{r}{n}}$$

where r is the rank of X. Equality holds when the non-zero eigenvalues of X are equal.

Thus, as the rank of a PSD matrix decreases its (maximum) angle with respect to the identity increases. The rank-1 matrices are the farthest from the identity, forming the fixed angle $\arccos(1/\sqrt{n})$ with respect to it.

Since the angle of a matrix in the PSD cone with respect to the central direction of the identity is apparently a parameter of importance, we seek a way of manipulating it. This ability is will be useful when we generate approximations to the PSD cone later. The following result gives a way of changing the angle of a matrix with respect to the identity that aids in this effort.

Result 4 (Angle change) *If X is an arbitrary symmetric matrix ($\neq I$) and α is a scalar then $\cos(X - \alpha I, I)$ is a monotonically decreasing function of α. That is, the angle between $X - \alpha I$ and I is a monotonically increasing function of α.*

The proof is in Appendix 2-C. By choosing α properly we may set this angle to whatever we desire. What is more important is that this function is monotonic. Since the angle of a matrix with respect to the identity matrix is determined by $\mathrm{Tr}(X)/(\|X\|_F\sqrt{n})$, the angle does not uniquely specify the matrix. In general there exist matrices of higher and lower rank with the same angle. What is useful to us is that for a *given* starting matrix it provides us with a way to change its position relative to the identity in a controlled way.

Support Hyperplanes

We characterize the support hyperplanes and halfspaces of the PSD cone next. A *hyperplane* is an $(n-1)$-dimensional affine subspace of an n-dimensional space [45]. Analytically, a hyperplane $\mathcal{H}(N,d)$ is the set of points $\mathcal{H}(N,d) = \{X|\langle X,N\rangle = d\}$ for some N and scalar d. The vector N is the *normal* to the hyperplane and d is the distance of the hyperplane from the origin. The set of points $\mathcal{HS}(N,d) = \{X|\langle X,N\rangle \geq d\}$ is the closed *halfspace* defined by the hyperplane $\mathcal{H}$. Note that, by convention, we are associating a halfspace with the hyperplane whose normal points *into* the halfspace. A closed halfspace $\mathcal{HS}$ is a (proper) support halfspace of a closed set K if $K \subset \mathcal{HS}$ and $\mathcal{H} \cap K \neq \emptyset$, where $\mathcal{H}$ is the hyperplane associated with $\mathcal{HS}$. A

hyperplane $\mathcal{H}$ is a (proper) support hyperplane of a closed set K if $\mathcal{H} \cap K \neq \emptyset$ but $\mathcal{H} \cap \mathsf{ri}\, K = \emptyset$, where ri denotes the relative interior of a set [45], i.e. the interior in the span of its elements. Analytically then, a hyperplane is a support hyperplane of K if $\langle Y, N \rangle \geq d$ for all $Y \in K$ with equality for some Y.

With these definitions we now characterize the support hyperplanes of the PSD cone. These hyperplanes are equivalently defined by their corresponding allowable pairs (N, d).

Result 5 (Support hyperplanes) *The support hyperplanes $\mathcal{H}(N, d)$ and corresponding support halfspaces of the PSD cone are precisely those with $d = 0$ and with a normal N which itself is a PSD matrix. Further, when N corresponds to a positive definite matrix the associated hyperplane only supports the PSD cone at the origin.*

The proof is in Appendix 2-D. Perhaps surprisingly then, the normals to the support hyperplanes of the PSD cone are members of the cone themselves. Further, the normals of the non-trivial hyperplanes (those supporting the cone at some place other than its point) lie on the cone's boundary, since they are associated with the PSD matrices of rank less than n. This result can actually be seen to be a consequence of the cone being self dual. The situation is illustrated for the $n = 2$ case in Figure 2-3.

Now a convex set K with non-empty interior is smooth at a boundary point p if p belongs to no more than one support hyperplane of K [46, 47]. Clearly, the PSD cone is not smooth at its tip, which belongs to all the support hyperplanes. Perhaps surprising however is the following result:

Result 6 (Smooth points) *The only boundary points at which the cone of $n \times n$ PSD matrices is smooth are those corresponding to rank-$(n-1)$ PSD matrices.*

The proof of this result is given in Appendix 2-E. Note, in particular, that only for the $n = 2$ case is the surface of the PSD cone smooth (except for its tip, of course), since in this case alone the entire boundary is composed of matrices of rank $(n-1) = 1$. This case is also the only one we can visualize. For all other cases, for example the set of 3×3 PSD matrices, the boundary is not smooth. This non-smoothness is apparently responsible for the difficulty that has been noted in doing optimization over this set [48].

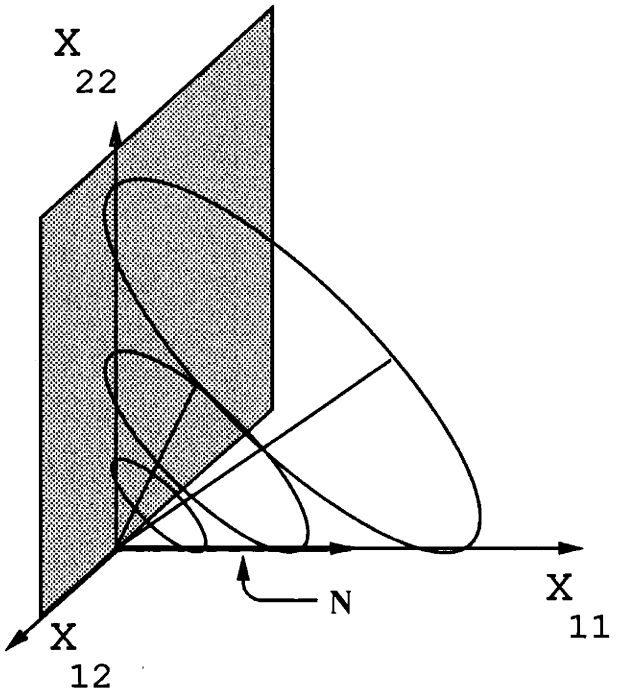

Figure 2-3: Illustration of support hyperplanes

Exterior Representation

It is a result of convex set theory that a non-empty closed convex set can be obtained as the intersection of its support halfspaces [45]. This prescription yields an "exterior" representation of any convex set and specifically of the PSD cone. Combining Result 5 and this representation we obtain the following characterization of the PSD set:

$$\mathrm{PSD}_n = \bigcap_{N \in \bigcup_{r=1}^{n} \mathrm{PSD}_n^{(r)}} \mathcal{HS}(N, 0)$$

Unfortunately, the cardinality of the set of (normalized) PSD matrices is infinite, implying that there is an infinite number of support hyperplanes to the full PSD cone (via the normals). From Result 5 we may even throw out those hyperplanes with positive definite normals as being redundant (since they only support the cone at its tip, the origin), but we are still left with an infinite number of semi-definite matrices as normals. Actually, not even all the semi-definite normals are needed to characterize the PSD set. We have the following result:

Result 7 (PSD characterization) *Let $1 \leq r \leq n$ be fixed, then*

$$PSD_n = \bigcap_{N \in PSD_n^{(r)}} \mathcal{HS}(N, 0)$$

The proof is in Appendix 2-F. Only the singular PSD normals of fixed rank are therefore actually needed to describe the PSD cone. In particular, we may describe the cone using only the set of rank-1 normals. This is the smallest such set, in the sense that it has the smallest dimension, n. This observation immediately gives the following corollary.

Corollary 2

$$PSD_n = \bigcap_{N \in PSD_n^{(1)}} \mathcal{HS}(N, 0)$$

Interestingly we may obtain this corollary directly from the classical result given as condition (2) of Result 2. Analytically, this result is just a restatement of Lemma 1 in the original space, but the interpretation is completely different.

Even though we have reduced the set of support hyperplanes required to describe the PSD cone to only those with rank-1 PSD normals, an infinite number of them is still needed for an exact characterization of the cone. With a *finite* such support set, however, we may *approximate* the PSD cone "from the outside" by using the resulting halfspaces to generate a polyhedral bound for the set. This approach is a useful one, and we discuss it in more detail later.

Rank-1 Normal Geometry

Since the rank-1 PSD matrices appear to be central to representing the PSD cone, their geometry is now briefly discussed. We saw above that the rank-1 PSD matrices are also normals to the support hyperplanes of the PSD set. We know that these matrices are on the boundary of the PSD cone, but what else can we say about them? Our central result is the following.

Result 8 (Rank-1 geometry) *The rank-1 PSD matrices of unit norm are contained in the hyperplane $\mathcal{H}(\frac{1}{\sqrt{n}}I, \frac{1}{\sqrt{n}})$. Further, in this hyperplane, these matrices lie on an $(n^2+n-2)/2$ ball of radius $\sqrt{(n-1)/n}$ centered at $1/n\, I$.*

Note that this hyperplane is an $(n-1)(n+2)/2$-dimensional affine subspace. The proof is in Appendix 2-G.

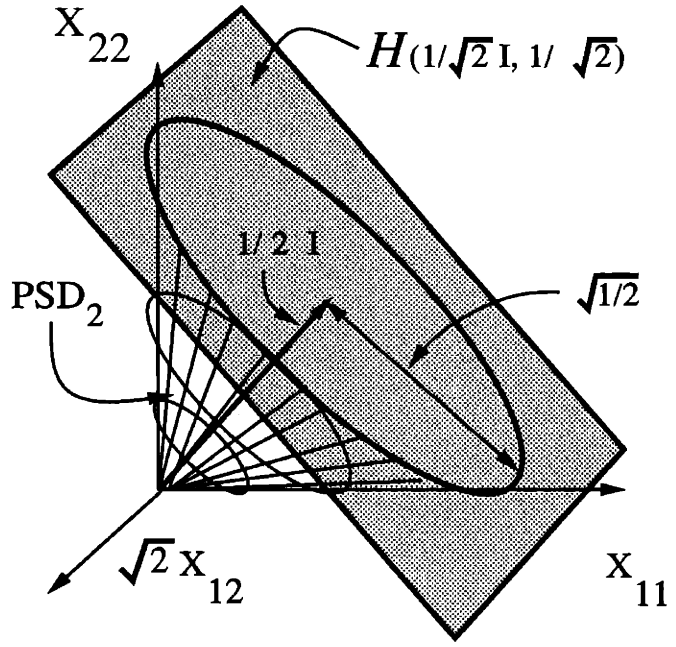

Figure 2-4: Illustration of Rank-1 matrix geometry

Any rank-1 PSD matrix of unit norm can be written as uu^T, where u is a unit n-vector. Since these vectors u have a smooth parameterization on the unit ball in R^n, we see that the unit rank 1 PSD matrices must actually be contained on an $(n-1)$-dimensional manifold. From the above result, this manifold is apparently not flat, i.e. confined to an $(n-1)$-dimensional affine subspace. We note that we have not detected any of the above special structure for the rank $r > 1$ PSD matrix case.

As an example of the above observations, consider the $n = 2$ case. This case is shown in Figure 2-4 in the vector space S. Here the coordinates are $[X_{11}\ \sqrt{2}X_{12}\ X_{22}]^T$. In this case the rank-1 matrices correspond to the boundary of the PSD cone. From the result, the points corresponding to the unit rank-1 matrices are constrained to lie in a hyperplane. For this case the containing hyperplane is simply the plane shown in the figure. The result also says these points will lie on a ball in this hyperplane of radius $\sqrt{1/2}$. This ball is the circle drawn on the plane in the figure. Finally, we know the points are confined to a 1-D manifold. This manifold coincides with the circle in this case. Note that for the $n = 2$ case this manifold *is* flat. For the case $n = 3$, the underlying space is 6-dimensional, so we cannot even draw it. For this case, the hyperplane would be 5-dimensional, the ball surface 4-dimensional, and the manifold 2-dimensional. This illustrates the difficulty in working with these sets even

for the physically motivated case of 3×3 matrices.

Facial Structure

The facial structure of a convex set (and a cone) is an important determiner of its geometry [49], and we examine this structure for the PSD cone next. The general theory is reviewed and then applied to the case of the PSD cone. Our review is based on [45, 40, 39]. Let K be a closed convex set. A convex subset F of K is called a face if $x, y \in K$ and $(x + y) \in F$ imply $x \in F$. The subsets $\emptyset$ and K are both faces of K called the improper faces. All other faces are called proper faces. The following theorem confirms our intuitive notion of what a face should be:

Theorem 1 ([45]) *If F is a face of a closed convex set K in R^d, with $F \neq K$, then $F \subset \mathsf{rb}\, K$.*

where $\mathsf{rb}\, K$ denotes the relative boundary of K. This theorem shows that the faces of a set are extreme subsets of the set associated with its boundary. What is more, these subsets form natural building blocks for the set. We have the following result:

Theorem 2 ([45]) *Let K be a closed convex set in R^d. For each $x \in K$ there is a unique face F of K such that $x \in \mathsf{ri}\, F$.*

where, again, $\mathsf{ri}\, F$ denotes the relative interior of F. In particular, the proper faces of K can have no points in $\mathsf{ri}\, K$, the interior of the set itself.

With the above intuition, we now give the following characterization of the faces of the PSD cone from [39]:

Theorem 3 ([39]) *Every face of the PSD cone is of the form:*

$$F_{\mathcal{S}} = \left\{ X \in PSD_n \,|\, X^{(i)} \perp \mathcal{S}, i = 1, \ldots, n \right\}$$

where $\mathcal{S}$ is a subspace of R^n and $X^{(i)}$ denotes the i-th column of X. Conversely, given a subspace $\mathcal{S} \subseteq R^n$, $F_{\mathcal{S}}$ is a face of the PSD cone.

From this result we can see that each face of the cone of $n \times n$ PSD matrices is isomorphic to the cone of $r \times r$ PSD matrices for some $0 \leq r \leq n$ [39]. Thus, the proper faces (and boundary) of the PSD cone are composed of those elements of the PSD set that are of less than full rank. This implies that the faces of the PSD cone must have dimension $r(r + 1)/2$ for some $0 \leq r \leq n$. Not all dimensions are represented!

There are certain special faces of a convex set. First we discuss those classes depending on the *dimension* of the face. One such special type of face is termed a *facet*. The facets of a convex set K are the faces of K with dimension 1 less than the dimension of K (and are thus like hyperplanes). This definition of facets extends our intuitive notion of the faces of a polyhedron. From the previous theorem and the arguments following it, is interesting to note that the PSD cone has no facets!

Another special type of face is termed an *extreme ray*. These faces are the 1-dimensional faces, and extend our notion of an edge. As an equivalent characterization of these faces, note that a ray of a convex cone is extreme if it can be deleted from the cone with the resulting cone remaining convex [38]. A point of an extreme ray is termed an *extremal*. It is a result of cone theory that a convex cone is the convex hull of its extreme rays [40]. For the PSD cone, the extremals are precisely the rank-1 matrices [39, 38]. Thus (as we know from linear algebra) every positive semi-definite matrix can be obtained as a convex combination of rank-1 PSD matrices. This yields an "interior" representation of the cone [45], dual to the exterior one given above.

Now a cone in an n-dimensional vector space is termed *polyhedral* if it has a finite set of extreme rays [40]. Unfortunately, since the cardinality of the set of rank-1 PSD matrices is infinite, an infinite number of them is needed to characterize the full PSD cone, so it is *not* polyhedral, as mentioned before. Again, we may approximate the cone, this time from the "inside," by using a finite set of extremals. We discuss such ideas later. It is interesting to note that we may obtain the PSD cone from the rank-1 matrices in *two* completely different ways: via an interior representation using a convex combination of rank-1 PSD matrices or from an exterior representation, using the halfspaces generated by rank-1 matrix normals.

Finally, the zero-dimensional faces are termed the *extreme points* of a convex set. Note that an extreme point is a face while an extremal is just a point on an extreme ray, and thus not necessarily a face. From the definition of a face it follows that a cone has no (proper) extreme points. Specifically, the PSD cone has no proper extreme points. To see this, consider a cone and any (non-zero) point X in it. Now $X = X/2 + X/2$, and $X/2$ is in the cone but clearly $X \neq X/2$, so X cannot be a (zero-dimensional) face. The extreme points of a set coincide with our notion of vertices. The problem is that cones, being unbounded, are not compact. In fact, we have the following result:

Theorem 4 ([47]) *If K is a non-empty compact set in R^d then K has at least one extreme point.*

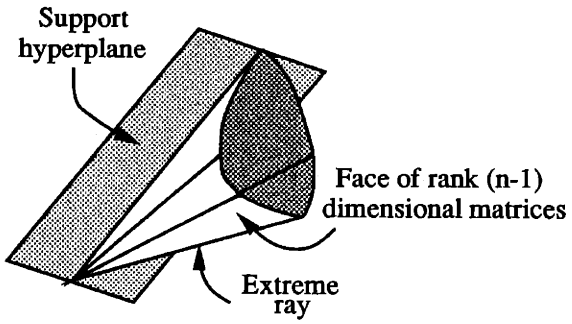

Figure 2-5: Schematic view of PSD cone

Even further, for a *compact* convex set we have the following result, known in different guises as Minkowski's Theorem [45] or the Krein-Milman Theorem [47]:

Theorem 5 ([47]) *If K is a compact convex set in R^d then K is the convex hull of its extreme points.*

Exposed Faces

To finish our consideration of the facial structure of the PSD cone, we examine a special group of faces of a set termed the *exposed* faces. A face F of a closed convex set K is a (proper) exposed face if $F = \mathcal{H} \cap K$ for some (proper) support hyperplane $\mathcal{H}$. A point $X \in K$ is called an *exposed point* if $\{X\}$ is an exposed face. The exposed faces are thus closely connected with the support hyperplanes of a cone. The exposed points are also extreme points. The PSD set therefore has none. As to the exposed faces, we have the following result:

Result 9 (Exposed faces) *Every proper face of the PSD cone is exposed.*

The proof is in Appendix 2-H. In summary, we give an overall schematic sense of the PSD cone in Figure 2-5.

2.4 Approximating the PSD Cone

While the cone of positive semi-definite matrices has much structure, it poses several challenges for optimization. The cone is not polyhedral, i.e. finitely generated, or even smooth (except for the $n = 2$ case) and the positive semi-definite conditions

are quite non-linear in the matrix elements. This non-linearity makes their incorporation into estimation algorithms difficult, generally requiring iterative non-linear approaches [8]. We now consider approximations to the PSD cone with the aim of remaining within the linear estimation framework that is developed in the next chapter. We seek conditions leading to *linear* or *quadratic* inequality constraints which may be easily combined with (3.6) to yield a linear least-squares problem subject to the derived linear or quadratic constraints. Such problems have been widely studied and efficient finitely terminating algorithms exist for their solution. We are also interested in iterative techniques using extensions of existing approaches or exploiting the special structure of the PSD cone we have exposed above. We begin by demonstrating the existence of arbitrarily close polyhedral approximations to the cone. With this theoretical basis established, we proceed to develop both interior and exterior approximations to the cone.

2.4.1 Existence of Approximations

Here we demonstrate the existence of approximating polyhedrons to the PSD cone. We accomplish this by truncating or capping the PSD cone to obtain a compact set that is adequate for our purposes. We may then invoke approximation results of convex set theory to achieve the desired existence statement.

It is a result of convex set theory [47] that for *compact* convex sets, there exist polyhedral approximations that approach them arbitrarily closely. Formally

Theorem 6 ([47]) *If K is a compact convex body in R^n, then for each $\epsilon > 0$ there exists a convex polyhedron P such that*

$$P \subset K \subset P_\epsilon$$

where

$$P_\epsilon = \bigcap_{p \in P} \mathcal{B}(p, \epsilon)$$

and $\mathcal{B}(p, \epsilon)$ is the ball of radius ϵ centered at p.

The PSD cone is already convex and closed, and by "capping" the cone we can obtain a bounded set. Using our knowledge of the geometry of the PSD cone, we accomplish this capping simply by intersecting the PSD cone with the halfspace that has $-I$ as its normal. This matches our intuitive notion of capping the cone. Analytically

we consider the set $\widehat{\mathrm{PSD}}_n(d) = \mathrm{PSD}_n \cap \mathcal{HS}(-I/n, d)$, where d is the distance of the bounding hyperplane from the origin. We have the following result:

Result 10 (Capped cone) *The capped cone $\widehat{PSD}_n(d) = PSD_n \cap \mathcal{HS}(-I/n, d)$ is a compact set.*

The proof is given in Appendix 2-I. This result coupled with the previous theorem yields the following corollary on the existence of polyhedral approximations to $\widehat{\mathrm{PSD}}_n(d)$.

Corollary 3 (Polyhedral Approximations) *The capped cone $\widehat{PSD}_n(d)$ may be approximated arbitrarily closely by a polyhedron.*

Clearly the sets $\widehat{\mathrm{PSD}}_n(d)$ monotonically nest as a function of d, the distance of the capping hyperplane from the origin. For any real PSD problem the solution will always lie within some know maximum range. Thus we may always choose d large enough so that the solution confined to $\widehat{\mathrm{PSD}}_n(d)$ will be the same as the solution confined to PSD_n. Thus capping the cone is conceptually not a real limitation.

2.4.2 Exterior Approximations

Exterior approximations of the PSD cone are presented here. By an exterior approximation we mean a set that contains the PSD cone, and is thus "exterior" to the PSD set. Later we consider interior approximations of the PSD cone, which are wholly contained within the PSD set. The approximations considered here are characterized by the fact that there exist elements of the approximating set that are *not* PSD matrices.

Finite Support

We start by using the observations of Section 2.3.1. Perhaps the simplest exterior approximation to the PSD set (or any convex set for that matter) is obtained by considering the intersection of a finite set of its support half-spaces $\mathcal{HS}(N_i, 0)$ where $N_i \in \mathrm{PSD}_n$. Coupling this observation with our insights into the geometry of the PSD cone yields the following lemma:

Lemma 2 *Given a finite set $\{N_i\}$ of (unit) PSD normals with $N_i \in PSD_n$, the associated external approximating set*

$$\mathcal{E}xt \equiv \bigcap_{N \in \{N_i\}} \mathcal{HS}(N, 0)$$

satisfies

$$\mathcal{E}xt \supset PSD_n$$

Further if $\mathcal{E}xt_k$ *is the set associated with* $\{N_i\}_k$, *then*

$$\{N_i\}_2 \supset \{N_i\}_1 \implies \mathcal{E}xt_2 \subseteq \mathcal{E}xt_1$$

Finally, since the PSD set is exactly obtained when $\{N_i\} = PSD_n^{(r)}$, *we have for each* $1 \leq r \leq n$ *that*

$$\bigcap_{\{N_i\}_k \subseteq PSD_n^{(r)}} \mathcal{E}xt_k = PSD_n$$

Thus the more PSD normals that are used the better the approximation. We also know that the normals corresponding to positive definite matrices support the PSD cone only at its point at the origin and are thus redundant. It is more efficient in general to choose PSD normals of rank less than n for our approximations.

This class of approximations is convenient for constrained optimization, and leads to linear inequality constraints. To see this fact, let $n_i = \Xi(N_i)$ be the representation of the matrix normals N_i with respect to the basis $\{M_\ell\}$ and similarly let $x = \Xi(X)$. The constraint that $X \in \mathcal{E}xt$ is then equivalent to the linear inequality constraint $[n_1|n_2|\cdots|n_q]^T x \geq 0$. Thus any optimization problem with a solution constrained to lie in $\mathcal{E}xt$ is equivalent to a problem in S_n that is constrained by linear inequalities.

Tessellations

The question remains of how to best choose the hyperplanes, or equivalently the normals to the hyperplanes, of the approximating set. From our previous work with the support hyperplanes of the PSD set above, we know that the smallest such group of hyperplane normals we may work with to obtain the PSD cone corresponds to the set of rank-1 PSD normals. Let us restrict our attention to approximating sets $\mathcal{E}xt$ obtained from such sets of rank-1 normals $\{N_i\}$. Now any unit rank-1 PSD matrix N may be written as $N = uu^T$, with $u \in R^n$ a unit vector. Note that u and $-u$ yield the same matrix N. Apart from this reflection, each unit rank-1 matrix is associated with a unique vector on the unit n-hemisphere, corresponding to the eigenvector of its non-zero eigenvalue. Conversely, any group of vectors on the unit n-hemisphere can be seen to generate a set of rank-1 unit PSD matrices. This observation yields the following result.

Result 11 (Support Tessellation) *Any tessellation of the unit n-hemisphere with vertices u_i induces an exterior approximation of PSD set as*

$$\mathcal{E}xt \equiv \bigcap_{N \in \{u_i u_i^T\}} \mathcal{HS}(N, 0)$$

Thus we may easily generate approximations to the PSD cone by simply choosing sets of points on the half n-sphere. Further, if $N_i = u_i u_i^T$ then $\cos(N_i, N_j) = \cos(u_i, u_j)^2$, so uniform spacing of the vectors u_i leads to uniform spacing of the normals N_i. An example for the $n = 2$ case is given in Figure 2-6. In (a) the PSD cone is shown together with an external approximation $\mathcal{E}xt$ using 4 faces. To the right is shown the intersection of the respective boundaries with the hyperplane $\mathcal{H}(I/n, 1)$. Note that we need at least $n(n+1)/2$ hyperplanes, and thus vectors u_i, if the intersection of the approximation with the hyperplane is to be a closed polyhedron. In (b) are shown the unit vectors u_i associated with $\mathcal{E}xt$.

Closeness Measure: Angle Bounds

We now give some insight and bounds on how good a particular external approximation of the form given in Result 11 is. Since the PSD cone and the approximation are both unbounded, the maximum distance between points in the approximation and those in the PSD cone is unbounded.

We take as our closeness measure the minimum value over the PSD cone of the maximum of the cosine of the *angle* between points of the approximation and those of the PSD cone. We maximize over points in the approximation and minimize over points in the PSD cone. Suppose E is an edge of the polyhedral cone $\mathcal{E}xt$. The angles of points in the approximation with any point of the PSD cone will have a maximum at some such edge. The cosine of the minimum achieved over all PSD points of this maximum angle is then our measure. This angle is shown in Figure 2-7 as θ and captures a notion of the difference between the approximation and the PSD cone.

Let us focus on an approximation edge E such as that shown in the figure. Given a unit edge matrix E of the approximation with at least one positive eigenvalue, the cosine of the minimum angle of this edge with the PSD set is given by:

$$\max_{X \in \mathbf{PSD_n}} \cos(X, E) = \sqrt{\sum_{\lambda_i(E) \geq 0} \lambda_i^2(E)}$$

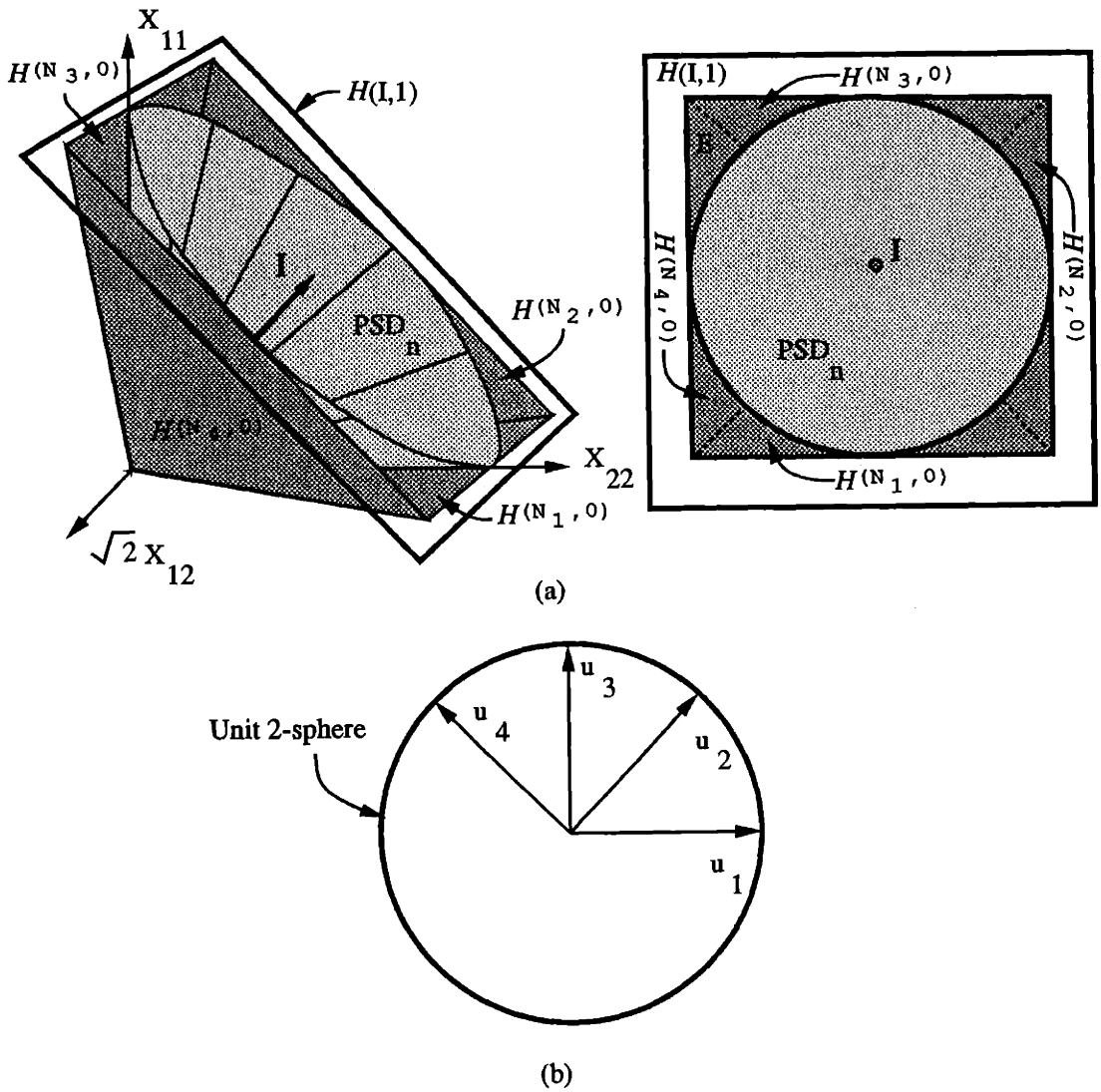

Figure 2-6: Exterior approximation example for $n = 2$

where $\lambda_i(E)$ are the eigenvalues of E. This expression is derived in Appendix 2-J. In practice, we would then find the minimum of this cosine value over the set of active approximation edges. The above cosine is larger, and hence the angle smaller and the approximation better, the closer E is to a PSD matrix (recall that if E is a unit matrix, then $\sum_i \lambda_i^2(E) = 1$). This matches our intuition.

Such an edge is obtained as the intersection of $n(n+1)/2 - 1$ of the support halfspaces. Specifically, for a given set of halfspaces $\{\mathcal{HS}(N_i, 0)\}$, E is obtained as

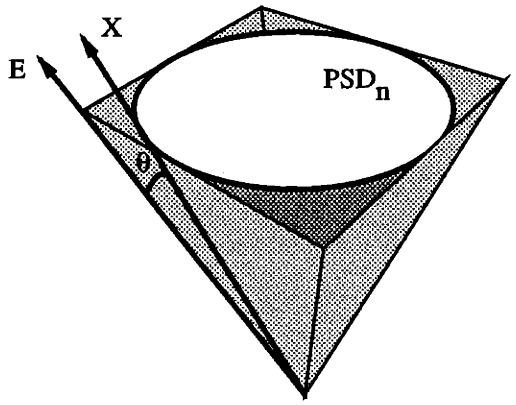

Figure 2-7: Measure of difference between approximation and PSD cone

the solution of the set of equations:

$$\langle N_i, E \rangle = 0, \qquad i = 1, 2, \ldots n(n+1)/2 - 1$$

with $\sum_i \lambda_i^2(E) = 1$. Thus finding an edge E from the normals N_i is essentially a problem in finding an element of a nullspace. As a result the detailed relationship between the N_i and E is difficult to make explicit. Still, we can see that when the N_i are close to each other, the resulting E will be close to a PSD matrix.

2.4.3 Interior Approximations

Now we consider *interior* approximations of the PSD cone. By interior, we mean that the approximating sets are contained completely within the PSD cone. We start by using the observations of Section 2.3.1, that the PSD set is convex and one-sided along the directions corresponding to the diagonal elements of the matrix. These properties comprise the generic structure of the PSD set. Following this treatment, we use more of the special structure of the PSD cone.

Convex Polyhedrons

Since the PSD set is a convex set, perhaps the simplest interior approximation to the set is to use a convex polyhedron whose vertices are contained within the set. The convex combination [45] of any finite set of PSD matrices yields such a polyhedron. Given a finite set of such "vertex" matrices $V_i \in \text{PSD}_n$, the corresponding interior

approximating set $\mathcal{I}nt$ based on them is given by:

$$\mathcal{I}nt = \left\{ \sum_{i=1}^{q} \alpha_i V_i \mid \sum_{i=1}^{q} \alpha_i = 1,\ \alpha_i \geq 0 \right\}$$

This class of approximations is again convenient for constrained optimization, leading once more to an inequality constraint on a certain transformed estimate. To see this fact, let $v_i = \Xi(V_i)$ be the representation of the vertex matrices V_i with respect to the basis $\{M_\ell\}$ and similarly let $x = \Xi(X)$. The constraint that $X \in \mathcal{I}nt$ is then equivalent to the statement that $X = \sum_{i=1}^{q} \bar{x}_i V_i$ for some $\bar{x}_i \geq 0$ with $\sum \bar{x}_i = 1$. In terms of x we have that

$$x = [v_1|v_2|\cdots|v_q]\bar{x} \tag{2.8}$$

where $\bar{x} = [\bar{x}_1|\bar{x}_2|\cdots|\bar{x}_q]^T$ and $\|\bar{x}\|_1 = 1$ with $\bar{x}$ confined to the first orthant. Thus in any optimization problem we may change variables using (2.8), then solve the resulting problem for $\bar{x}$ with the following inequality constraint:

$$\left[\begin{array}{cccc} \multicolumn{4}{c}{I} \\ \hline 1 & 1 & \cdots & 1 \\ -1 & -1 & \cdots & -1 \end{array}\right] \bar{x} \geq \left[\begin{array}{c} \bigcirc \\ \hline 1 \\ 1 \end{array}\right]$$

The first block of the constraint assures that $\bar{x}$ is confined to the first orthant ($\bar{x}_i \geq 0$) and the second block forces $\sum |\bar{x}_i| = \|\bar{x}\|_1 = 1$. Hence any optimization problem with a solution constrained to lie in $\mathcal{I}nt$ can be transformed into an equivalent linear inequality constrained problem on $\bar{x}$. We note that (with considerably more effort) we may also phrase this directly as a halfspace intersection problem in terms of x. To do this we must find the facets of the polytope induced by the vertex matrices V_i, and their associated normals. When $n = 2$ this operation is not too difficult but even for the $n = 3$ case (where the space is 6 dimensional), finding these facets is not trivial.

Coordinate Boxes

Of course there are certain special polyhedrons we may consider which simplify the constraint set. For example, we can consider boxes oriented along the coordinate axes of the basis elements $\{M_\ell\}$. For the standard basis we present in Appendix 2-A these directions correspond to perturbations of the individual matrix entries X_{ij}. These

sets are of the form

$$\left\{X \,|\, \underline{X}_{ij} \leq X_{ij} \leq \overline{X}_{ij}\right\}$$

Such a box defines a set of *interval matrices*. The notion of interval matrix considered here is defined directly in terms of the elements of the matrix X_{ij}. In Chapter 3 we consider another notion of interval matrix based on the PSD definition, where $X \geq \underline{X}$ means $(X - \underline{X})$ is a PSD matrix. Interval matrices have been studied with regard to their stability properties [50, 51, 52, 53, 54] but not, to our knowledge, for their properties of positive semi-definiteness. A set of (symmetric) interval matrices, being a box, will be in the PSD set if and only if its vertices are PSD matrices. This follows from the discussion so far. Further, however, since the boundary of the PSD set is one sided along directions corresponding to the diagonal elements, we actually do not need to check those vertices involving $\overline{X}_{ii}$. Equivalently, we may let $\overline{X}_{ii} \to \infty$ and the box will remain in the PSD set. Thus we lose no generality in considering boxes of the form:

$$\mathcal{I}nt = \left\{X \,|\, \underline{X}_{ij} \leq X_{ij} \leq \overline{X}_{ij},\ i \neq j,\ \ \underline{X}_{ii} \leq X_{ii}\right\} \tag{2.9}$$

This interval box will be in PSD_n if and only if the $2^{n(n-1)/2}$ elements of the set:

$$\left\{X \,|\, X_{ij} = \underline{X}_{ij},\ \text{ or }\ X_{ij} = \overline{X}_{ij},\ i \neq j,\ X_{ii} = \underline{X}_{ii}\right\}$$

are in PSD_n. These corners comprise the set of significant or binding corners.

Once such a box as (2.9), contained in the PSD set, has been defined, we would like to use it in optimization problems. Because of the special orientation of the box, finding the induced facets and their normals is straightforward. The set of support halfspaces $\mathcal{HS}(N, d)$ to (2.9) is given by (N, d) pairs of the form:

$$N_k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{matrix} \\ i \\ \\ \end{matrix}, \qquad d_k = \underline{X}_{ii}$$

$$N_k = \begin{bmatrix} 0 & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{matrix} i \\ j \\ \\ \end{matrix}, \qquad d_k = \sqrt{2}\underline{X}_{ij} \qquad i \neq j$$

$$\begin{matrix} & i & & j & \end{matrix}$$

$$N_k = \begin{bmatrix} 0 & -1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{matrix} i \\ j \\ \\ i \end{matrix} \quad \begin{matrix} \\ \\ \\ j \end{matrix}, \quad d_k = -\sqrt{2}\underline{X}_{ij} \qquad i \neq j$$

If we let $x = \Xi(X)$ be the representation of the matrix X with respect to our standard basis $\{M_\ell\}$, then the constraint that X is contained in the set (2.9) is equivalent to a simple linear inequality constraint. For the $n = 2$ case this constraint is given by:

$$\left[\begin{array}{ccc} & I & \\ \hline 0 & -1 & 0 \end{array}\right] x \geq \left[\begin{array}{c} \underline{X}_{11} \\ \sqrt{2}\underline{X}_{12} \\ \underline{X}_{22} \\ \hline -\sqrt{2}\overline{X}_{12} \end{array}\right]$$

where $\underline{X}_{ij}$ and $\overline{X}_{ij}$ are the boundaries of the set. For the $n = 3$ case the constraint becomes:

$$\left[\begin{array}{cccccc} & & & I & & \\ \hline 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{array}\right] x \geq \left[\begin{array}{c} \underline{X}_{11} \\ \sqrt{2}\underline{X}_{12} \\ \sqrt{2}\underline{X}_{13} \\ \underline{X}_{22} \\ \sqrt{2}\underline{X}_{23} \\ \underline{X}_{33} \\ \hline -\sqrt{2}\overline{X}_{12} \\ -\sqrt{2}\overline{X}_{13} \\ -\sqrt{2}\overline{X}_{23} \end{array}\right]$$

The higher order cases are similar. Thus any optimization problem with a solution constrained to lie in (2.9) is isomorphic to a linear inequality constrained problem. For the $n = 2$ case we show what the approximating region looks like in Figure 2-8 for a generic choice of $\underline{X}_{ij}$ and $\overline{X}_{ij}$.

Extreme Ray Approximation

As mentioned in our discussion of the facial structure of the PSD cone, we can obtain PSD_n as the positive linear combination of the (infinite) set of extreme rays of the cone. These are the PSD rank-1 matrices. A straightforward interior approximation

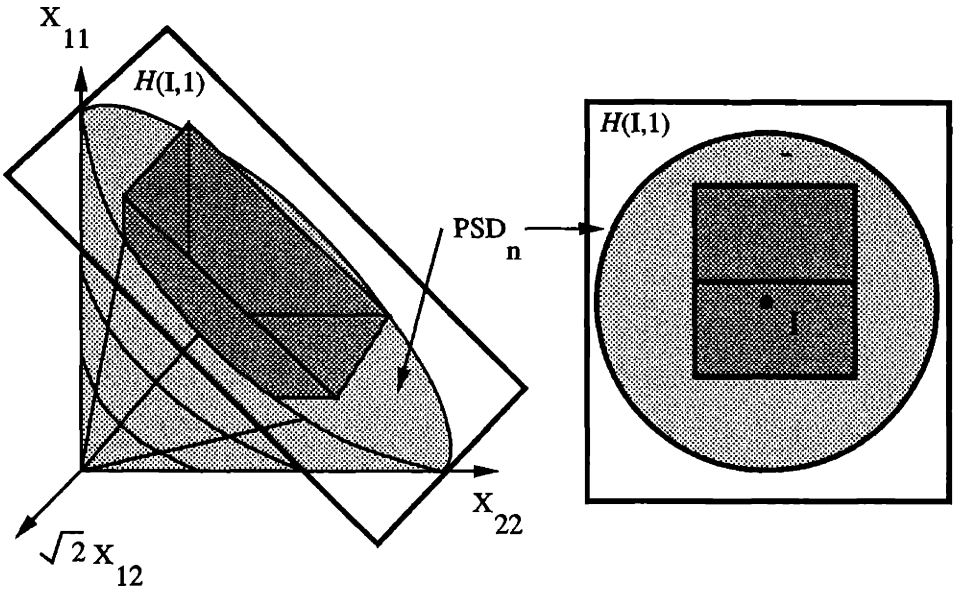

Figure 2-8: Interval approximation example for $n = 2$

then is to use the positive linear combination of a *finite* subset of these rank-1 matrices. This approach can be viewed as the dual of the approximation using a finite set of support hyperplanes, which we discussed above. We again choose a finite set of extreme rays $R_i \in \mathrm{PSD}_n^{(1)}$, but rather than using them as normals to hyperplanes we directly generate an interior approximation $\mathcal{I}nt$ based on them. Formally we have the following lemma:

Lemma 3 (Extreme Ray Approximation) *Given a finite set $\{R_i\}$ of (unit) PSD rays with $R_i \in PSD_n^{(1)}$, the associated internal approximating set*

$$\mathcal{I}nt \equiv \left\{ X \mid X = \sum_{i=1}^{q} \alpha_i R_i, \;\; \alpha_i \geq 0 \right\}$$

satisfies

$$\mathcal{I}nt \subset PSD_n$$

Further, if $\mathcal{I}nt_k$ is the set associated with $\{R_i\}_k$, then

$$\{R_i\}_2 \supset \{R_i\}_1 \implies \mathcal{I}nt_2 \supseteq \mathcal{I}nt_1$$

Finally, since the PSD set is exactly obtained when $\{R_i\} = PSD_n^{(1)}$, *we have that*

$$\bigcup_{\{R_i\}_k \subseteq PSD_n^{(1)}} \mathcal{I}nt_k = PSD_n$$

Thus, the more extreme rays we use in an interior approximation, the better the approximation will be, in general.

Again, this class of approximations is convenient for constrained optimization, leading to positivity constraints on a linearly transformed problem. To see this property, let $r_i = \Xi(R_i)$ be the representation of the extreme ray R_i with respect to the basis $\{M_\ell\}$ and similarly let $x = \Xi(X)$. The constraint of the lemma that $X \in \mathcal{I}nt$ is then equivalent to the statement that

$$x = [r_1|r_2|\cdots|r_q]\bar{x} \tag{2.10}$$

where now $\bar{x} = [\bar{x}_1|\bar{x}_2|\cdots|\bar{x}_q]^T$ and $\bar{x}$ is confined to the first orthant. This formulation is much like (2.8), except here there is no constraint on the norm of $\bar{x}$. In any optimization problem using this approximation, we may thus change variables using (2.10), then solve the resulting problem for $\bar{x}$ under a nonnegativity condition. In a least squares formulation, this results in a non-negative least squares (NNLS) problem for which finitely terminating algorithms exist [55]. As before, we note that we may also phrase these constraints directly as a halfspace intersection problem in terms of x (again with considerably more effort) .

Tessellations

As in the external representation case, the rank-1 unit PSD rays R may be written as $R = uu^T$, with $u \in R^n$ a unit vector. Thus, any group of vectors on the unit n-hemisphere can be seen to generate a set of rank-1 unit PSD extreme rays. This observation yields the following result, dual to Result 11.

Result 12 *Any tessellation of the unit n-hemisphere with vertices* u_i *induces an interior approximation of the PSD set as*

$$\mathcal{I}nt \equiv \left\{ X \,|\, X = \sum_{i=1}^{q} \alpha_i u_i u_i^T,\ \alpha_i \geq 0 \right\}$$

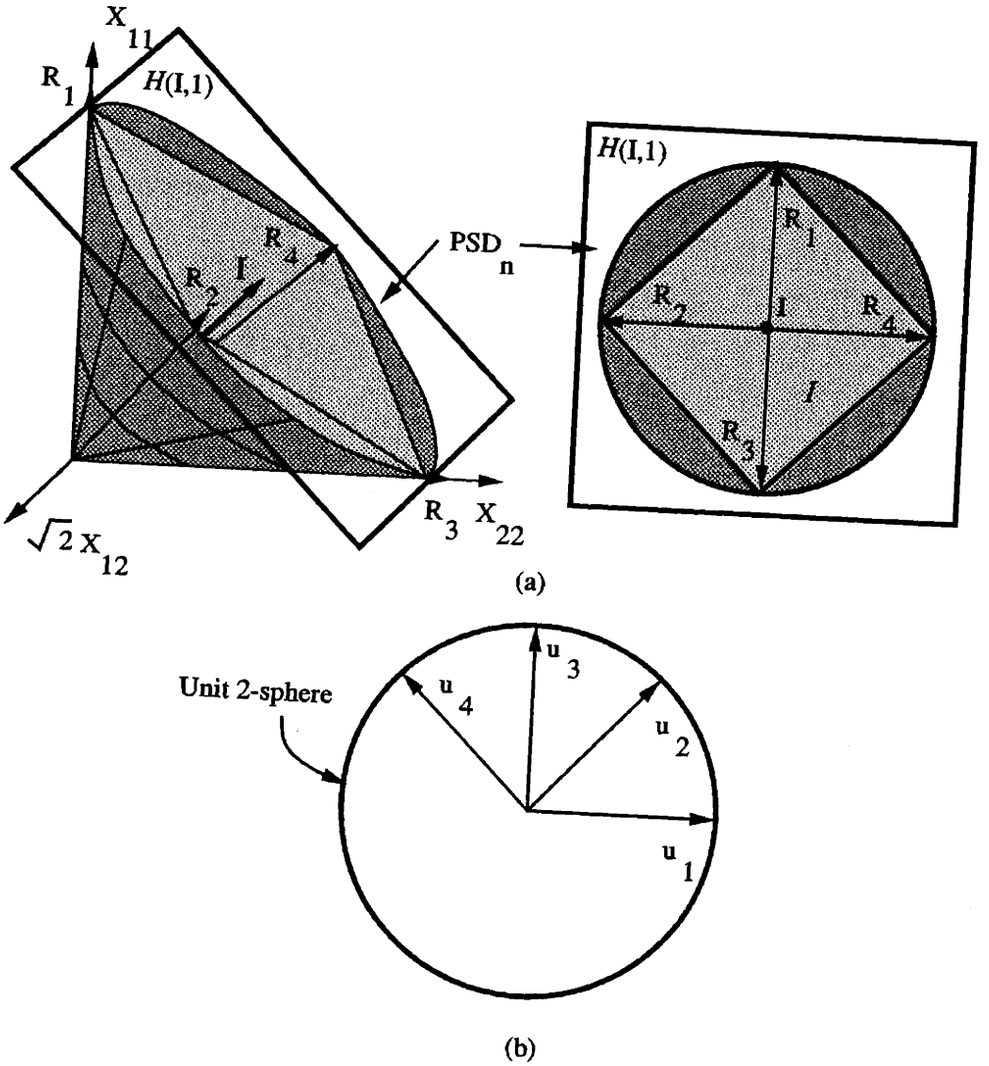

Figure 2-9: Extreme ray approximation example for $n = 2$

We may easily generate *interior* approximations to the PSD cone by simply choosing sets of points on the half n-sphere. Again, if $R_i = u_i u_i^T$ then $\cos(R_i, R_j) = \cos(u_i, u_j)^2$, and uniform spacing of the vectors u_i leads to uniform spacing of the rays R_i. An example for the $n = 2$ case is given in Figure 2-9. In (a) the PSD cone is shown together with an internal approximation $\mathcal{I}nt$ using the convex hull of 4 extreme rays. To the right is shown a slice through the respective set boundaries by the hyperplane $\mathcal{H}(I/n, 1)$. Note that we need at least $n(n+1)/2$ rays, and thus vectors u_i, if the intersection of the approximation with this hyperplane is to be a closed polyhedron. In (b) are shown the unit vectors u_i associated with the R_i.

Geršgorin Polyhedral Cone

Here we present a seemingly unrelated interior approximation derived from the Geršgorin Theorem. This purely algebraic result yields a simple and finite set of sufficient conditions for positive semi-definiteness of a matrix. We interpret the resulting approximation in the context presented above, showing how to view it as the intersection of a finite set of half-spaces, The theorem itself provides a simple bound on the location of the eigenvalues of a matrix in terms of an inequality based on the diagonal elements and the row sums of the matrix. Thus the approximation is actually based on the eigenvalue condition for positive semi-definiteness of a symmetric matrix. The underlying eigenvalue theorem is given by:

Theorem 7 (Geršgorin) *If $X = [X_{ij}]$ is an $n \times n$ matrix then every eigenvalue of X lies in at least one of the disks*

$$\{z \,|\, |z - X_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |X_{ij}|\} \qquad i = 1, 2, \ldots, n.$$

The theorem states that the eigenvalues of X are contained in the union of the disks centered on the diagonal elements and with radius equal to the respective (absolute value) row sums. The column sums can also be used. This leads us to a particularly simple, if combinatorial, sufficient condition for the positive semi-definiteness of a matrix. Specifically, a symmetric matrix will be positive definite if it satisfies

$$X_{ii} \geq \sum_{\substack{j=1 \\ k \neq j}}^{n} |X_{ij}| \qquad i = 1, 2, \ldots, n.$$

The subset of the space of symmetric matrices that satisfies this condition can be obtained as the intersection of a set of half spaces, thus producing a polyhedral approximation to the PSD cone. Since the condition is only sufficient, this polyhedral cone must be contained wholly within the PSD cone. This approximation gives us the following result:

Result 13 (Geršgorin Polyhedral Cone) *Consider the polyhedral cone defined by*

$$\mathcal{I}nt = \bigcap_{N \in \{N_{g_i}\}} \mathcal{HS}(N, 0)$$

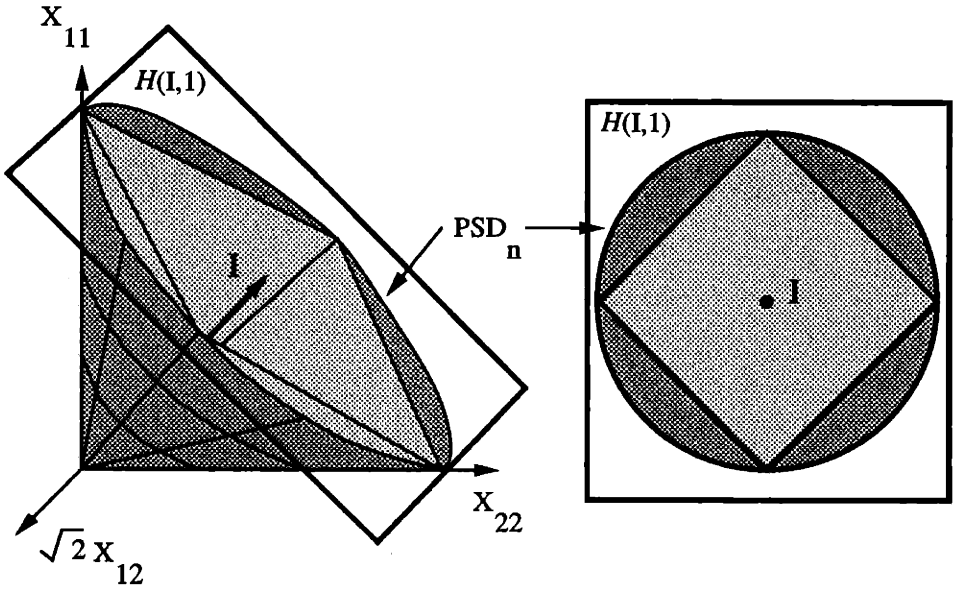

Figure 2-10: Geršgorin polyhedral cone for $n = 2$

where the set of "Geršgorin normals" $\{Ng_i\}$ is composed of all $n \times n$ matrices of the form

$$Ng_i = \frac{1}{\sqrt{n+1}} \begin{bmatrix} & & \pm 1 & & & \\ & \bigcirc & \pm 1 & & \bigcirc & \\ \pm 1 & \pm 1 & 2 & \pm 1 & \pm 1 & \pm 1 & \pm 1 \\ & & \pm 1 & & & \\ & \bigcirc & \pm 1 & & \bigcirc & \\ & & \pm 1 & & & \\ & & \pm 1 & & & \end{bmatrix}.$$

where the 2 is one of the diagonal elements and the sign pattern of the ± 1 terms is arbitrary (consistent with symmetry). This cone is contained in the PSD cone.

The resulting approximation is a cone defined by $n2^{(n-1)}$ halfspace intersections. For the $n = 2$ case only 4 hyperplanes or tests are obtained while for the $n = 3$ case there are 12 inequalities to satisfy. Figure 2-10 illustrates the polyhedral approximation for the $n = 2$ case. This figure is similar to Figure 2-8, only here the approximation is generated using hyperplanes rather than extreme rays. Also, here there is no freedom in the choice of the shape. The approximating cone is fixed by the theorem.

As with the previous approximations to the PSD cone we have presented, the polyhedral cone given in Result 13 is convenient for constrained optimization, since

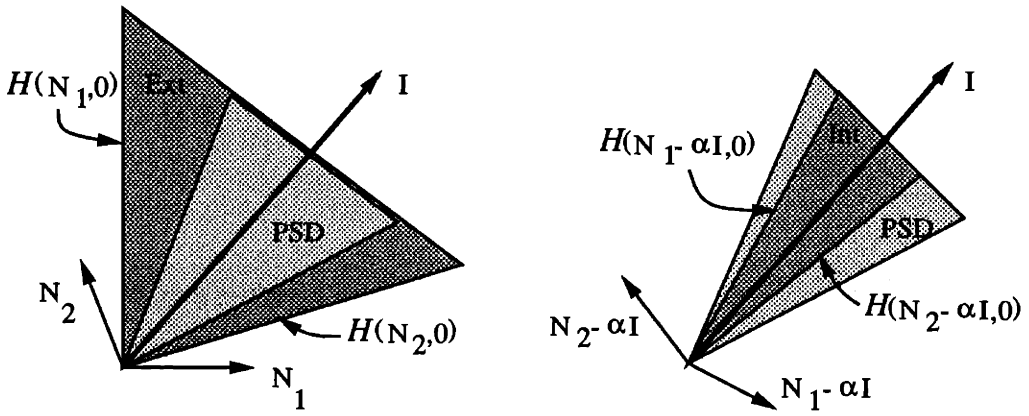

Figure 2-11: Illustration of perturbing hyperplane normals

it involves linear inequality constraints on the entries of the matrix. To see this fact, let $ng_i = \Xi(Ng_i)$ be the representation of the matrix normals Ng_i with respect to the basis $\{M_\ell\}$ and similarly let $x = \Xi(X)$. The constraint that $X \in \mathcal{I}nt$ of Result 13 is then equivalent to the linear inequality constraint $[ng_1|ng_2|\cdots|ng_q]^T x \geq 0$. Thus any optimization problem with a solution constrained to lie in $\mathcal{I}nt$ is equivalent to a linear inequality constrained problem in S_n.

Interior Approximations from Exterior Ones

We now show how to turn the exterior approximations of Lemma 2 into interior approximations. We accomplish this feat with the help of Result 4. Given an exterior approximation formed using a set of PSD rank-1 normals, we use this result to move the normals of the given (support) hyperplanes uniformly away from the central direction of the identity, with the effect of shrinking the associated polyhedral cone. With enough such shrinking the resulting cone will be completely contained in the PSD cone. This idea is illustrated in Figure 2-11.

We restrict our attention to the case of rank-1 normals, though the extension to the general case should be straightforward. Consider the family of approximating sets $\mathcal{F}(t)$, $t \in [0,1)$, defined as follows:

$$\mathcal{F}(t) \equiv \bigcap_{N \in \{N_i(t)\}} \mathcal{HS}(N,0)$$

where $\{N_i(t)\}$ is a set of q normals and for each $1 \leq i \leq q$

$$N_i(t) \equiv u_i u_i^T - \alpha_i(t) I$$

with $u_i \in R^n$ of unit 2 norm and $\alpha_i(t)$ is a smooth strictly monotonic function of t with $\alpha_i(0) = 0$ and $\alpha_i(1) = 1/n$. These sets can be seen to actually be a family of approximating polyhedral cones with $\mathcal{F}(0)$ equal to an exterior finite support hyperplane approximation. As t increases the exterior set closes, eventually becoming an interior approximation. As an example of the functions $\alpha_i(t)$ we might choose, we could pick $\alpha_i(t) = t/n$ to achieve uniform shrinking. We have the following result for the above family $\mathcal{F}(t)$, with $\mathsf{cone}^+(\cdot)$ representing the positive cone of the elements of the argument:

Result 14 *For any family of polyhedral cones $\mathcal{F}(t)$, $t \in [0,1)$, defined as above, if $\mathsf{cone}^+\{N_i(0)\}$ is full and its interior contains I ($I \in \mathsf{int}\ \mathsf{cone}^+\{N_i(0)\}$) then the following hold:*

1. $\mathcal{F}(0) \supset PSD_n$.
2. $\mathcal{F}(t_1) \supset \mathcal{F}(t_2)$ *if* $0 \leq t_1 < t_2 < 1$
3. $\bigcap_{t \in [0,1]} \mathcal{F}(t) = \mathsf{cone}^+(I) \subset PSD_n$.
4. *There exists a unique minimum $t^\star$ (and a corresponding $\alpha^\star$) such that $\mathcal{F}(t^\star) \subseteq PSD_n$.*

The proof of Result 14 is in Appendix 2-K. Note that the requirement that the positive cone of the set $\{N_i(0)\}$ is full implies that the number q of hyperplanes of the approximation is at least as large as the dimension of the space $n(n+1)/2$. The requirement that the central PSD direction I is in the *interior* of this cone is necessary to achieve part 4 of the result. The condition assures that the intersection of the approximation's bounding hyperplanes with the hyperplane $\mathcal{H}(I/n, 1)$ is a closed bounded region. Part 3 essentially says that the approximations converge to the identity direction.

These results provide a way of generating arbitrary interior or exterior approximations to the PSD cone. Any tessellation of the unit half-sphere induces an exterior approximation which we may then convert to an interior approximation (or anything in between) via the above procedure. Consider the $n = 2$ case again with uniformly

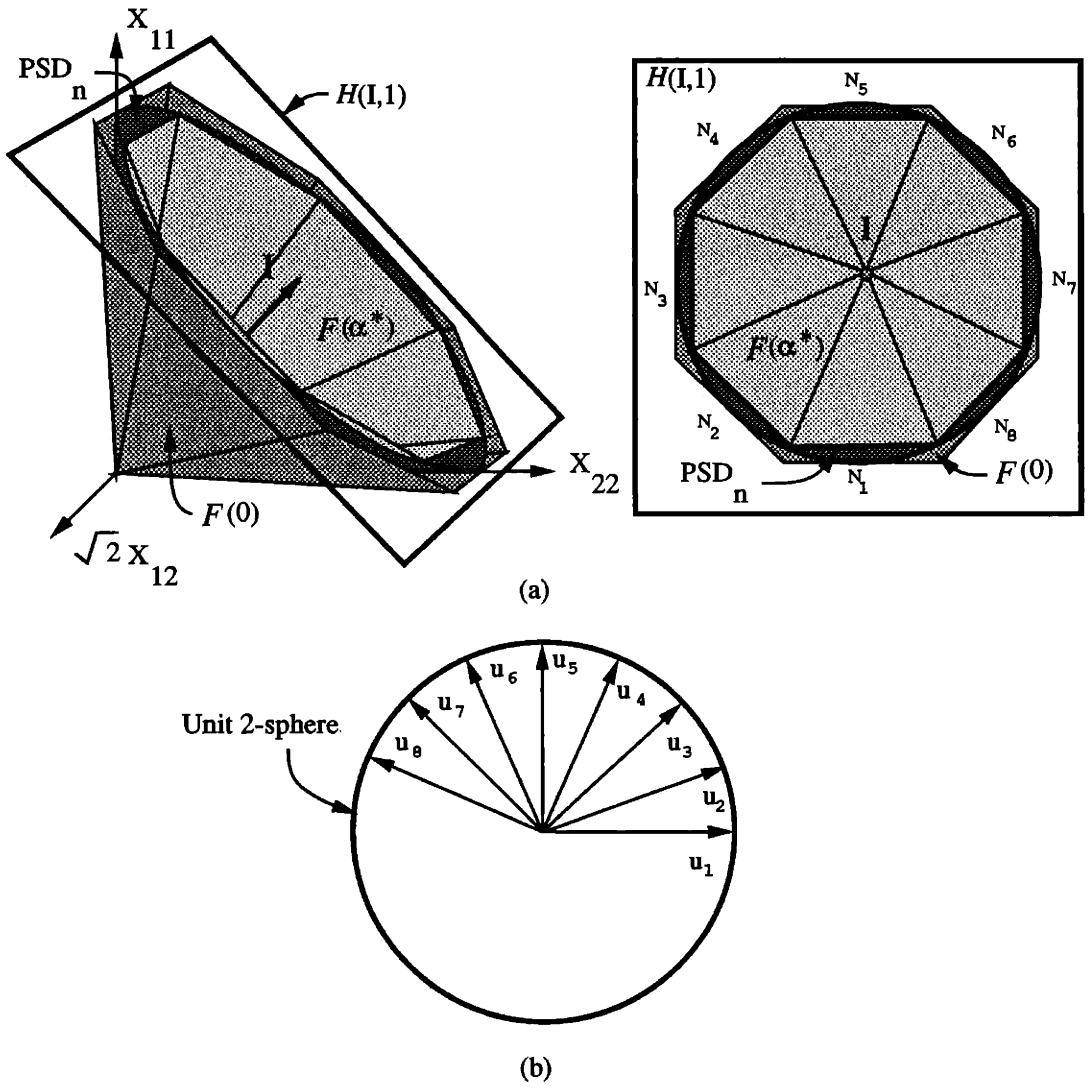

Figure 2-12: Regular polyhedral cones.

spaced normals $N_i(0)$, corresponding to uniformly spaced vectors u_i on the half circle. The resulting interior cones have regular polyhedrons for their cross sections. An example with 8 faces is shown in Figure 2-12. In (a) the PSD cone is shown together with the approximations corresponding to the initial ($\alpha = 0$) and final ($\alpha = \alpha^\star$) values of $\alpha(t)$ while in (b) the corresponding vectors u_i are shown. We can calculate the minimum $\alpha^\star$ given the $N_i(0)$ (equivalently the u_i) for this $n = 2$ case. If q is the number of faces of the regular approximation, then the minimum α required of each facet (they are the same by symmetry) to take the exterior support-based approximation to an interior approximation is given by $\alpha^\star = (1 - \cos(\pi/q))/2$. A log-log plot

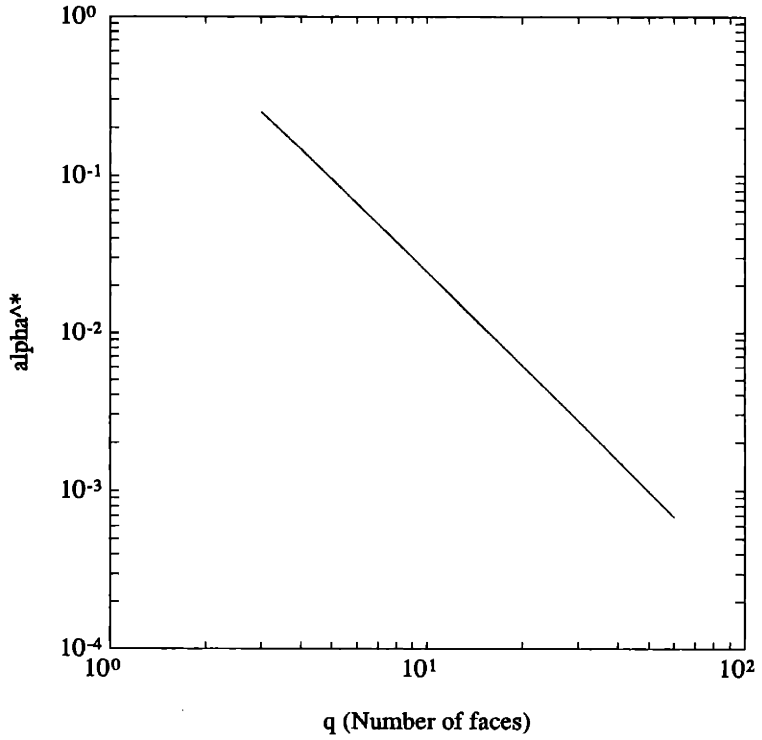

Figure 2-13: $\alpha^\star$ vs q

of $\alpha^\star$ vs q is given in Figure 2-13 for interest.

In general, finding $\alpha^\star$ for a particular problem requires the equivalent of numerical root finding. For a given α we can find the edge matrices of the resulting polyhedral approximation and test if they are all positive semi-definite or not. Finding $\alpha^\star$ then reduces to finding the "zero crossing" of this function, i.e. the transition of the edge set from indefinite to all positive semi-definite. As the dimension of the problem grows, the problem of finding the edges becomes prohibitive. For example, when $n = 3$ with q normals, there are ${}_qC_5$ potential intersections to check. If $q = 20$ this results in over 15,000 combinations to check. For an optimization problem with a fixed choice of $\{N_i(0)\}$ we would only have to perform this operation once, of course, and the approximation would then be determined using only the 20 faces and the determined value of $\alpha^\star$. For *any* α the resulting polyhedral cone leads to a set of linear inequality constraints for inclusion in the set. This form of constraint is once more convenient for optimization.

2.4.4 Summary

In summary, our goal was to find simple approximations to the conditions imposed by the constraint of positive semi-definiteness of a symmetric matrix. We have presented a number of different such approximations to the PSD cone, both algebraically and geometrically based. One group of approximations yields exterior approximations to the PSD cone, and includes the PSD cone in their interior along with other non-PSD elements. Another group is contained wholly within the PSD cone itself, producing interior approximations. We have also shown how to convert exterior approximations based on support hyperplanes to interior approximations, establishing the connection between the two in the process. A range of approximations exists between these two extremes, of course, with sets that are neither contained in nor contain the PSD set. Our investigation of the PSD cone has served to illuminate and unify these different approaches to approximating the PSD cone within a natural geometric framework using the structure of the PSD cone itself.

All of the approximations discussed above have in common the property that they are *finite* representations and that they lead, either directly or indirectly through a transformation, to a set of *linear inequality constraints* on the elements of the underlying matrix. Geometrically, these approximations yield finitely generated polyhedral or polyhedral cone sets in the space of the matrix entries. As we see in Chapter 3, this is convenient for purposes of optimization.

2.5 Conclusions

In this chapter we have examined a particular physically motivated symmetric projection mapping. We began by introducing the symmetric projection mapping in (2.3). The existence the of an isometric isomorphism between this mapping and an equivalent convenient linear matrix formulation was shown. This isomorphism greatly simplified certain aspects of the problem and appears to be an useful tool in approaching these problems.

Many of the problems using a symmetric matrix in which we are interested actually require a positive semi-definite matrix. Motivated by estimation problems requiring such a constraint, in Section 2.3 the geometry of the PSD cone was investigated. We provided a complete characterization of the support hyperplanes of the PSD cone, leading to a minimal support halfspace based representation of the cone. In addition we exposed the non-smooth nature of the PSD cone, and provided characterizations

of various other elements of its facial and extremal structure. These insights enabled a unified view of the overall geometry of the PSD cone and allowing us to generate approximating sets to the cone next.

Approximations to the PSD cone which were both contained in and contained the PSD cone were developed. These approximations all shared the common characteristic that they were defined at some level by a set of *linear inequality constraints* on the elements of the underlying matrix. These approximations included boxes oriented along the coordinate directions, Geršgorin polyhedral cones, external support halfspace cones, and interior extreme ray sets. A procedure was also presented to smoothly convert any support based exterior approximation into an interior approximation, generating a family of approximations in the process. Two complementary ways that tessellations of the unit half-sphere induce approximations of the PSD set, one exterior and one interior, were provided. These results are put to use in Chapter 3 to develop constrained matrix reconstruction.

2-A Symmetric space properties

Consider the vector space S_n induced on the set of $n \times n$ symmetric matrices by the inner product $\langle A, B \rangle_n \equiv \mathsf{Tr}(A^T B)$. Here we gather some results pertaining to this space and work through some examples. We define the standard basis on S_n to be $\{M_\ell^{(n)} \mid 1 \leq \ell \leq n(n+1)/2\}$ where

$$M_\ell^{(n)} = \begin{cases} e_i e_i^T & \text{if } \ell = (i-1)(2n+2-i)/2 + 1 \quad i = 1 \ldots n \\ \frac{e_i e_j^T + e_j e_i^T}{2} & \text{if } \ell = i(2n+1-i)/2 - n + j \quad 1 \leq i < j \leq n \end{cases} \tag{2.11}$$

and e_i is the i-th standard basis vector composed of all zeros except for a 1 in the i-th location.

Example

Both to fix notions and for its later utility, we work through the case when $m = 1$ and $n = 2$ (often corresponding to the case of a 1-dimensional projection of a 2-dimensional object). In equation (2.3), this means X is 2×2, Y a scalar, and A a 2-vector. For this case, our standard symmetric basis consists of the following three elements:

$$M_1^n = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad M_2^n = \begin{bmatrix} 0 & 1/\sqrt{2} \\ 1/\sqrt{2} & 0 \end{bmatrix}, \quad M_3^n = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

If the ij-th entry of X is X_{ij}, and the i-th entry of A is a_i, then we have the following:

$$\begin{aligned} Y &= A^T X A \\ &= [a_1 \;\; a_2] \begin{bmatrix} X_{11} & X_{12} \\ X_{12} & X_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \\ &= a_1^2 X_{11} + 2 a_1 a_2 X_{12} + a_2^2 X_{22} \end{aligned}$$

Now with respect to our standard basis, for x we obtain

$$\Xi(X) = x = \begin{bmatrix} X_{11} \\ \sqrt{2}\, X_{12} \\ X_{22} \end{bmatrix}$$

To obtain $\tilde{A}$ we apply the mapping Γ of Definition 2 yielding:

$$\tilde{A} = \Gamma(A) = \begin{bmatrix} a_1^2 & \sqrt{2}\, a_1 a_2 & a_2^2 \end{bmatrix}$$

Being a scalar, the representation of Y is just Y itself. It is easy to verify for this case that

$$Y = \tilde{A}x.$$

2-B Properties of the mapping Γ

Consider the mapping Γ relating (2.4) to (2.3) defined as in Chapter 2 by:

$$A \overset{\Gamma}{\longmapsto} \tilde{A} \tag{2.12}$$

where $\tilde{A}_{ij} = \langle M_i^{(m)}, A^T M_j^{(n)} A \rangle_m$. We will now prove the following relationships between A and $\tilde{A} = \Gamma(A)$.

Result 15 *Given the matrix A, if $\tilde{A}$ is the image of A under Γ, so that $A \overset{\Gamma}{\longmapsto} \tilde{A}$, then the following relations hold:*

For square A:

$$A \text{ idempotent} \implies \tilde{A} \text{ idempotent} \tag{2.13}$$

$$A \text{ orthogonal} \implies \tilde{A} \text{ orthogonal} \tag{2.14}$$

$$A_2 = A^T \implies \tilde{A}_2 = \tilde{A}^T \tag{2.15}$$

$$A \text{ symmetric} \implies \tilde{A} \text{ symmetric} \tag{2.16}$$

$$A \text{ projector} \implies \tilde{A} \text{ projector} \tag{2.17}$$

$$A^T v_i = v_i \lambda_i \implies \tilde{A}\tilde{v}_{ij} = \tilde{v}_{ij}\tilde{\lambda}_{ij} \tag{2.18}$$

For non-square A:

$$A^T A = I \implies \tilde{A}\tilde{A}^T = I \tag{2.19}$$

$$A^T \nu_i = \mu_i \sigma_i \implies \tilde{A}\tilde{\nu}_{ij} = \tilde{\mu}_{ij}\tilde{\sigma}_{ij} \tag{2.20}$$

where $\tilde{v}_{ij}$ is the representation of $(v_i v_j^T + v_j v_i^T)$ with respect to $\{M_\ell^{(m)}\}$, $\tilde{\lambda}_{ij} = \lambda_i \lambda_j$, ν_i and μ_i are the left and right singular vectors associated with the singular value σ_i of A, $\tilde{\nu}_{ij}$ is the representation of $(\nu_i \nu_j^T + \nu_j \nu_i^T)$ with respect to $\{M_\ell^{(n)}\}$, $\tilde{\mu}_{ij}$ is the representation of $(\mu_i \mu_j^T + \mu_j \mu_i^T)$ with respect to $\{M_\ell^{(m)}\}$ and $\tilde{\sigma}_{ij} = \sigma_i \sigma_j$ for $1 \leq i \leq$

$j \leq n$.

Note that the relations in (2.18) and (2.20) provide the relationship between the eigenstructure or singular structure of A and $\tilde{A}$. The eigenstructure relationship given above is known (in a less compact form) from considerations of the Kronecker product [35] but we have not seen the corresponding singular structure relationship.

Proof: To show (2.13) note that if A is idempotent then, for any X, $A^T A^T X A A = A^T X A$. Now if $B = A^T X A$ then $b = \tilde{A}x$ as defined above. But $B = A^T X A = A^T A^T X A A = A^T B A$, so $b = \tilde{A}b$ or $\tilde{A}x = \tilde{A}\tilde{A}x$, and (2.13) is proved.

To show (2.14), note that $\tilde{A}$ orthogonal implies orthonormal columns. Now using the definitions above, the inner product of column i and column j of $\tilde{A}$ is given by:

$$\begin{aligned} \langle A^T M_i^{(n)} A, A^T M_j^{(n)} A \rangle_m &= \mathsf{Tr}(A^T M_i^{(n)} A A^T M_j^{(n)} A) \\ &= \mathsf{Tr}(A^T M_i^{(n)} M_j^{(n)} A) \\ &= \mathsf{Tr}(M_i^{(n)} M_j^{(n)}) \\ &= \langle M_i^{(n)}, M_j^{(n)} \rangle_m \\ &= \delta_{ij} \end{aligned} \tag{2.21}$$

We now prove relationship (2.15). Let $A \overset{\Gamma}{\longmapsto} \tilde{A}_1$ and $A^T \overset{\Gamma}{\longmapsto} \tilde{A}_2$. We need to show that $\tilde{A}_{1_{ij}} = \tilde{A}_{2_{ji}}$. We have:

$$\begin{aligned} \tilde{A}_{1_{ij}} &= \langle M_i^{(m)}, A^T M_j^{(n)} A \rangle_m \\ &= \mathsf{Tr}(M_i^{(m)} A^T M_j^{(n)} A) \\ &= \mathsf{Tr}(A M_i^{(m)} A^T M_j^{(n)}) \\ &= \langle A M_i^{(m)} A^T, M_j^{(n)} \rangle_n \\ &= \langle M_j^{(n)}, A M_i^{(m)} A^T \rangle_n \\ &= \tilde{A}_{2_{ji}} \end{aligned}$$

where the third equality follows from $\mathsf{Tr}(PQ) = \mathsf{Tr}(QP)$. Now (2.16) follows from (2.15), since if $A \overset{\Gamma}{\longmapsto} \tilde{A}$ then $A^T \overset{\Gamma}{\longmapsto} \tilde{A}^T$, but $A = A^T$ so $\tilde{A} = \tilde{A}^T$. Finally, property (2.17) now follows from (2.13) and (2.16) and the fact that if A is a projector, then by definition $AA = A$ and $A^T = A$.

To show (2.18) we use the fact that the symmetric matrix represented by $\tilde{A}x$ is simply obtained as $A^T X A$, where X is the matrix represented by the vector x. Thus

the matrix represented by $\tilde{A}\tilde{v}_{ij}$ is just,

$$\begin{aligned} A^T(v_i v_j^T + v_j v_i^T)A &= \lambda_i\lambda_j v_i v_j^T + \lambda_j\lambda_i v_j v_i^T \\ &= \lambda_i\lambda_j(v_i v_j^T + v_j v_i^T) \\ &= \tilde{\lambda}_{ij}(v_i v_j^T + v_j v_i^T) \end{aligned}$$

Now, by definition, the representation of $(v_i v_j^T + v_j v_i^T)$ with respect to $\{M_\ell^{(m)}\}$ is $\tilde{v}_{ij}$, so combining these pieces we have shown that $\tilde{A}\tilde{v}_{ij} = \tilde{v}_{ij}\tilde{\lambda}_{ij}$ for $1 \leq i \leq j \leq n$. Relation (2.20) is shown similarly to (2.18).

To show (2.19), first note that each row of $\tilde{A}$ is the representation of a symmetric matrix of the form $c_{ij}(A_i A_j^T + A_j A_i^T)$, for $1 \leq i \leq j \leq m$, where A_i is column i of A and

$$c_{ij} = \begin{cases} 1 & \text{if } i = j \\ 1/\sqrt{2} & \text{otherwise} \end{cases}$$

Thus the inner product of two rows of $\tilde{A}$ is given by:

$$c_{ij}c_{k\ell}\,\mathsf{Tr}((A_i A_j^T + A_j A_i^T)(A_k A_\ell^T + A_\ell A_k^T)) = \begin{cases} 1 & \text{if } i = k \text{ and } j = \ell \\ 0 & \text{otherwise} \end{cases}$$

But $i = k$ and $j = \ell$ only when the two rows are the same. Thus $\tilde{A}\tilde{A}^T = I$, as desired. All relations are now shown. □

2-C Proof of Result 4

We show here that $f(\alpha) = \cos(X - \alpha, I)$ is a monotonically decreasing function of α. We do this by showing that $df/d\alpha$ is always negative. We have that

$$\begin{aligned} f(\alpha) &= \cos(X - \alpha, I) \\ &= \frac{\mathsf{Tr}(X - \alpha I)}{\|X - \alpha I\|_F \sqrt{n}} \\ &= \frac{\mathsf{Tr}(X) - \alpha n}{\sqrt{\alpha^2 n^2 - 2\alpha n \mathsf{Tr}(X) + n\|X\|_F^2}} \end{aligned}$$

Now the derivative of this last expression with respect to α is

$$\frac{df(\alpha)}{d\alpha} = \frac{-(\mathrm{Tr}(X)^2 + n\|X\|_F^2)}{\sqrt{n}\|X - \alpha I\|_F^3}$$

Since the denominator of this expression is always non-negative and the terms in the numerator are similarly non-negative, this shows that $f(\alpha) < 0$ for all α.

2-D Proof of Result 5

A hyperplane $\mathcal{H}(N, d)$ is a support hyperplane of PSD_n if and only if the following two conditions are satisfied:

$$\forall X \in \mathrm{PSD}_n \quad \langle N, X \rangle \geq d \tag{2.22}$$

$$\exists X \in \mathrm{PSD}_n \quad \langle N, X \rangle = d \tag{2.23}$$

First we show that $d = 0$ for any support hyperplane of PSD_n. First suppose there is a support hyperplane with $d \neq 0$. Let $X_0 \in \mathrm{PSD}_n$ be such that condition (2.23) is satisfied for this hyperplane. Now, ϵX_0 is also a PSD matrix for all $\epsilon > 0$, and $\langle N, \epsilon X_0 \rangle = \epsilon d$. As long as $d \neq 0$, we may always choose an $\epsilon > 0$ so ϵX_0 is a PSD matrix yet (2.24) is violated, regardless of the sign of d. Thus we must have $d = 0$.

Now we shall show that the normals N of the support hyperplanes of the PSD cone are themselves PSD matrices. For any symmetric matrix normal N, we may write N as:

$$N = \sum_{i=1}^{q} \lambda_i \, u_i u_i^T$$

where λ_i are the q non-zero eigenvalues of N and u_i are the corresponding eigenvectors. Since $\langle uu^T, X \rangle = u^T X u$ we have that

$$\langle N, X \rangle = \sum_{i=1}^{q} \lambda_i \, u_i^T X u_i$$

Combining this observation with the fact that $d = 0$, the conditions that a hyperplane must satisfy to be a support hyperplane become:

$$\forall X \in \mathrm{PSD}_n \quad \sum_{i=1}^{q} \lambda_i \, u_i^T X u_i \geq 0 \tag{2.24}$$

$$\exists X \in \text{PSD}_n \qquad \sum_{i=1}^{q} \lambda_i \, u_i^T X u_i = 0 \tag{2.25}$$

where as above, u_i, λ_i are the eigenvector/value pairs for the associated normal N. Now (2.24) is only satisfied for all PSD X if $\lambda_i \geq 0$ (if any $\lambda_i < 0$ consider the PSD matrix $u_i u_i^T$). Thus every supporting hyperplane normal corresponds to a PSD matrix.

Next we demonstrate that every PSD matrix is the normal of some supporting hyperplane of the cone. We only have to show that (2.24) and (2.25) are satisfied. If N is a PSD matrix, then $\lambda_i \geq 0$, thus (2.24) is trivially satisfied. Consider two cases. If N is semi-definite so at least one $\lambda_i = 0$, then (2.25) is satisfied for any $X = \sum \alpha_i u_i u_i^T$, where the sum is over indicies coresponding to the zero eigenvalues of N and $\alpha_i \geq 0$. If N is positive definite, so that all $\lambda_i > 0$, then the only PSD matrix satisfying (2.25) is the null matrix. Thus for this case the PSD cone is supported only at its point at the origin.

2-E Proof of Result 6

A boundary point is smooth if and only if it belongs to only one support hyperplane. For the PSD matrix X, let $r = \mathsf{rank}(X)$. We may write any rank r PSD matrix X as

$$X = \sum_{i=1}^{r} \lambda_i \, u_i u_i^T$$

where $u_i^T u_j = \delta(i,j)$ and $\lambda > 0$. Now the support hyperplanes of the PSD cone are $\mathcal{H}(N,0)$, where $N \in \text{PSD}_n$. Since N itself is a (unit) PSD matrix we may write it as

$$N = \sum_{j=1}^{q} v_j v_j^T$$

where $v_i^T v_j = \delta(i,j)$ and $1 \leq q \leq n$. Let us now characterize all the support hyperplanes to which a given X could belong by characterizing the corresponding set of v_j. A hyperplane will contain X if and only if

$$\begin{aligned} 0 &= \langle N, X \rangle \\ &= \mathsf{Tr}\left[\sum_{i=1}^{r} \sum_{j=1}^{q} \lambda_i \, (v_j v_j^T)(u_i u_i^T) \right] \end{aligned}$$

$$= \sum_{i=1}^{r}\sum_{j=1}^{q} \lambda_i (v_j^T u_i)^2$$

Since $\lambda_i > 0$, this can only be if $v_j \perp u_i$ for all i, j. Let $\mathcal{U}$ be the space spanned by the u_i. Let $\mathcal{U}^\perp$ be the orthogonal complement of $\mathcal{U}$, so $\dim(\mathcal{U}^\perp) = (n-r)$. The condition for the inclusion of X in a hyperplane is then that $v_j \in \mathcal{U}^\perp$. In addition, since the v_j are orthonormal, they must form an orthonormal basis for $\mathcal{U}^\perp$.

Let $\{u_j^\perp\}$ be an othonormal basis for $\mathcal{U}^\perp$. When $r = \mathsf{rank}(X) = (n-1)$ then $\dim(\mathcal{U}^\perp) = (n-r) = 1$ and thus there is only a single $v_j = u_1^{\perp T}$ that satisfies the condition $v_j \in \mathcal{U}^\perp$. Thus the only supporting hyperplane containing X has the normal $N = u_1^\perp u_1^{\perp T}$. In particular, only a single such hyperplane exists. When $r = \mathsf{rank}(X) < (n-1)$, however, then $\dim(\mathcal{U}^\perp) = (n-r) > 1$. Thus more than a single choice exists for N. For example the normals $N_1 = u_1^\perp u_1^{\perp T}$ and $N_2 = u_2^\perp u_2^{\perp T}$ both contain X. In fact any convex combination of N_1 and N_2 also contains X. Thus the result is proved.

2-F Proof of Result 7

We prove the result by showing that for a fixed $1 \le r \le n$

$$X \in \mathrm{PSD}_n \iff \langle X, N\rangle \ge 0, \ \forall N \in \mathrm{PSD}^{(r)}$$

Now let u_i, λ_i be the eigenvector/eigenvalue pairs of N. Since $\langle X, u_i u_i^T\rangle = u_i^T X u_i$, an equivalent statement is the following:

$$X \in \mathrm{PSD}_n \iff \sum_{i=1}^{r} \lambda_i \, u_i^T X u_i \ge 0, \ \forall u_i \in R^n, \ \lambda_i > 0 \tag{2.26}$$

Clearly, if X is a PSD matrix, then the right hand side is true. To show the converse we prove the contrapositive; that $X \notin \mathrm{PSD}_n$ implies the right hand side is false. For any symmetric X we may write:

$$X = \sum_{j=1}^{n} \rho_j \, v_j v_j^T$$

where ρ_j, v_j are the eigenvalue/eigenvector pairs of X. We assume that the eigenvectors are normalized and that the eigenvalues are ordered so that $\rho_1 \le \rho_2 \ldots$. Assume

X is not positive semi-definite so that $\rho_1 < 0$. Suppose in (2.26) that we choose $u_i = v_i$ for $i = 1, \ldots, r$. Then

$$\begin{aligned}\sum_{i=1}^{r} \lambda_i \, u_i^T X u_i &= \sum_{i=1}^{r} \lambda_i \sum_{j=1}^{n} \rho_j \, v_i^T (v_j v_j^T) v_i \\ &= \lambda_1 \rho_1 + \sum_{i=2}^{r} \lambda_i \rho_i \\ &\leq \lambda_1 \rho_1 + \sum_{i=2}^{r} \lambda_i |\rho_i|\end{aligned}$$

where $\lambda_i > 0$. Now if we choose

$$\lambda_1 > \frac{\sum_{i=2}^{r} |\lambda_i \rho_i|}{|\rho_1|} > 0$$

then expression above is negative, since ρ_1 is negative. We have therefore constructed an $N = \sum_{i=1}^{r} \lambda_i v_i v_i^T$ which is positive semi-definite of rank r with $\langle X, N \rangle < 0$ and the contrapositive is proved.

2-G Proof of Result 8

By definition $X \in \mathcal{H}(I/\sqrt{n}, 1/\sqrt{n})$ means that $\langle X, I/\sqrt{n} \rangle = 1/\sqrt{n}$. If X is a unit rank-1 matrix, then $X = uu^T$ for some unit $u \in R^n$. Now

$$\begin{aligned}\langle X, I/\sqrt{n} \rangle &= \frac{1}{\sqrt{n}} \langle uu^T, I \rangle \\ &= \frac{1}{\sqrt{n}} u^T u \\ &= 1/\sqrt{n}\end{aligned}$$

so the first part of the result is shown.

To show the second part we must show that $\|X - I/n\|_F^2 = (n-1)/n$, when $X = uu^T$ for some unit vector $u \in R^n$. Now

$$\begin{aligned}\|uu^T - I/n\|_F^2 &= \mathsf{Tr}\left[(uu^T - I/n)(uu^T - I/n)\right] \\ &= \mathsf{Tr}\left[uu^T - (2/n)\, uu^T + I/n^2\right] \\ &= (n-2)/n + 1/n\end{aligned}$$

$$= (n-1)/n$$

and the result is shown.

2-H Proof of Result 9

Suppose S is arbitrary but fixed. We show that the face associated with $\mathcal{S}$ can be obtained as the intersection of a hyperplane and the PSD set and is thus exposed. First recall that the proper faces of the PSD cone are of the form:

$$\left\{X \mid S^T X = 0\right\}$$

for some $S = [s_1|s_2|..|s_r]$, $1 \leq r \leq (n-1)$, where the columns s_i of S form an orthonormal basis for an r-dimensional subspace $\mathcal{S}$ of R^n. We may parameterize any X in the PSD set as:

$$X = \sum_{i=1}^{q} \lambda_i \, u_i u_i^T$$

where $\lambda_i > 0$, $1 \leq q \leq n$ and $u_i \in R^n$, with $u_i \perp U_j$. Now applying this parameterization to the face associated with S we obtain for the face:

$$\left\{X \mid S^T \sum_{i=1}^{q} \lambda_i \, u_i u_i^T = 0\right\}$$

or equivalently

$$\left\{X \mid \sum_{i=1}^{q} \lambda_i (s_j^T u_i) \, u_i = 0, \;\; 1 \leq j \leq r\right\}$$

Since the u_i are orthogonal, this can only be if $u_i \perp s_j$. Thus the face associated with S is given by the set:

$$F_{\mathcal{S}} = \left\{X = \sum_{i=1}^{q} \lambda_i \, u_i u_i^T \mid \lambda_i > 0, \; u_i \perp s_j, \; \forall i, j\right\}$$

Now consider the hyperplane $\mathcal{H}(SS^T, 0)$. We shall show that the intersection of this hyperplane and the PSD set yields the set $F_{\mathcal{S}}$. Now the set $\mathcal{H}(SS^T, 0) \cap \mathrm{PSD}_n$ is given by the following:

$$\left\{X \mid \langle SS^T, X \rangle = 0, \; X \in \mathrm{PSD}_n\right\}$$

Using the parameterization of elements of the PSD set given above, this becomes

$$\left\{ X = \sum_{i=1}^{q} \lambda_i \, u_i u_i^T \mid \langle SS^T, X \rangle = 0, \; \lambda_i > 0, \; u_i \in R^n \right\}$$

or equivalently

$$\left\{ X = \sum_{i=1}^{q} \lambda_i \, u_i u_i^T \mid \sum_{j=1}^{r} \sum_{i=1}^{q} \lambda_i \, (s_j^T u_i)^2 = 0, \; \lambda_i > 0, \; u_i \in R^n \right\}.$$

Again, this can only be if $u_i \perp s_j$. Thus, the intersection of $\mathcal{H}(SS^T, 0)$ and the PSD set is given by:

$$\mathcal{H}(SS^T, 0) \bigcap \mathrm{PSD}_n = \left\{ X = \sum_{i=1}^{q} \lambda_i \, u_i u_i^T \mid \lambda_i > 0, \; u_i \perp s_j \; \forall i, j \right\}$$

But this is just $F_{\mathcal{S}}$. Since the subspace $\mathcal{S}$ was arbitrary, the result is proved.

2-I Proof of Result 10

Here we shall prove that the capped PSD cone $\widehat{\mathrm{PSD}}_n(d) = \mathrm{PSD}_n \cap \mathcal{HS}(-I/n, d)$ is a compact set. We do this by showing that the set is bounded. Since the resulting set is closed and bounded it is compact. Our goal is to show that $\|X\|_F^2$ is finite, where $X \in \mathrm{PSD}_n$ and $X \in \mathcal{HS}(-I/n, d)$. Note that $X \in \mathcal{HS}(-I/n, d)$ is equivalent to $\mathsf{Tr}(X) \leq nd$. For a symmetric matrix

$$\|X\|_F^2 = \sum_{i=1}^{n} \lambda_i^2$$

where $\lambda_i \geq 0$ are the eigenvalues of X. Now we have

$$\mathsf{Tr}(X) = \sum_{i=1}^{n} \lambda_i \leq nd$$

Thus

$$0 \leq \sum_{i=1}^{n} \lambda_i \leq nd$$

so

$$\sum_{i=1}^{n} \lambda_i^2 \leq (\sum_{i=1}^{n} \lambda_i)^2 \leq (nd)^2$$

Thus

$$\|X\|_F^2 = \sum_{i=1}^{n} \lambda_i^2 \leq (nd)^2$$

and the result is shown.

2-J Angle Bounds on External Approximation

For a given edge of an external approximation, here we find a bound on the maximum of the cosine of the angle between this edge and the PSD cone. If E is a (unit) edge vector, then clearly the PSD matrix X_E whose angle is smallest with E is given by the closest PSD matrix to E. As we show in equation (3.13), this matrix is obtained as

$$X_E = \sum_{\lambda_i(E) \geq 0} \lambda_i(E)\, v_i v_i^T$$

where $\lambda_i(E)$ are the eigenvalues of E and v_i are the associated eigenvectors. Now for the cosine of the angle between X_E and E we obtain:

$$\begin{aligned} \cos(E, X_E) &= \frac{\mathsf{Tr}(E, X_E)}{\|E\|_F \|X_E\|_F} \\ &= \frac{\mathsf{Tr}[(\sum_{\lambda_i(E)\geq 0} \lambda_i v_i v_i^T)(\sum_j \lambda_j v_j v_j^T)]}{\sqrt{\sum_{\lambda_i(E)\geq 0} \lambda_i^2}} \\ &= \sqrt{\sum_{\lambda_i(E)\geq 0} \lambda_i^2} \end{aligned}$$

which is the desired expression.

2-K Proof of Result 14

Part 1

Part 1 of the result follows easily from the definition of the family $\mathcal{F}(t)$ and Result 11.

Part 2

To show part 2 we shall need two lemmas. The first lemma shows that the hypothesis that $I \in \text{int } \mathsf{cone}^+\{N_i(t)\}|_{t=0}$ holds for any t. This lemma is then used to prove the second lemma, which shows that all elements of $\mathcal{F}(t)$ form an acute angle with the

identity for all $t \in [0,1)$.

Lemma 4 *For the normals $N_i(t)$ defined in Result 14, I is in the interior of the positive cone of $\{N_i(t)\}$ for all $t \in [0,1)$.*

Proof of lemma: By hypothesis, I is in the interior of the positive cone of $\{N_i(0)\}$, thus we may write I as $I = \sum_{i=1}^{q} c_i(0)N_i(0)$ with $c_i(0) > 0$. Futher, $\sum_{i=1}^{q} c_i(0) = n$ since

$$n = \langle I, I \rangle = \left\langle I, \sum_{i=1}^{q} c_i(0)N_i(0) \right\rangle = \sum_{i=1}^{q} c_i(0)\,\mathsf{Tr}\,(u_i u_i^T) = \sum_{i=1}^{q} c_i(0).$$

Now $N_i(t) = N_i(0) - \alpha_i(t)I$ so $N_i(0) = N_i(t) + \alpha_i(t)I$. Substituting in the sum for I and rearranging terms we have that $I = \sum_{i=1}^{q} c_i(t)N_i(t)$ where

$$c_i(t) = \frac{c_i(0)}{1 - \sum_{j=1}^{q} \alpha_i(t)c_i(0)}.$$

Now $1/n > \alpha(t) \geq 0$ for $1 > t \geq 0$, thus for the denominator we have that

$$1 - \sum_{j=1}^{q} \alpha_i(t)c_i(0) > 1 - \sum_{j=1}^{q} c_i(0)/n = 0.$$

Hence $c_i(t) > 0$ for each i, and I can be written as a positive linear combination of the $N_i(t)$ for all $0 \leq t < 1$. Note that when all the functions $\alpha_i(t) = \alpha(t)$ then

$$\sum_{i=1}^{q} c_i(t) = \frac{n}{1 - n\alpha(t)}$$

This proves the first lemma.

The second lemma shows the elements of $\mathcal{F}(t)$ form an acute angle with the identity for any $t \in [0,1)$.

Lemma 5 *For any $X \in \mathcal{F}(t)$, $t \in [0,1)$, $\langle X, I \rangle > 0$ (i.e. $\mathsf{Tr}\,(X) > 0$).*

Proof of lemma: From the first lemma, for any t in the above range we may write $I = \sum_{i=1}^{q} c_i(t)N_i(t)$, with $c_i(t) > 0$. Thus:

$$\langle X, I \rangle = \left\langle X, \sum_{i=1}^{q} c_i(t)N_i(t) \right\rangle = \sum_{i=1}^{q} c_i(t)\,\langle X, N_i(t) \rangle.$$

Since $X \in \mathcal{F}(t)$ we have that $\langle X, N_i(t)\rangle \geq 0$ for each i. If follows that $\langle X, I\rangle \geq 0$. To show strict inequality we proceed as follows. The interior of $\mathsf{cone}^+\{N_i(t)\}$ is non-empty by hypothesis, so the set $\{N_i(t)\}$ must span the underlying space. This observation implies that $\langle X, N_i(t)\rangle = 0$ for all i if and only if $X = 0$, which it is not. Taken together, these observations imply that at least one term $\langle X, N_i(t)\rangle$ in the above sum must be greater than zero. Since $c_i(t) > 0$, the lemma is shown.

To prove the main result, let $X \in \mathcal{F}(t_2)$, then for each i

$$\langle X, N_i(0) - \alpha(t_2)I\rangle \geq 0$$

or

$$\langle X, N_i(0)\rangle \geq \alpha(t_2)\langle X, I\rangle > \alpha(t_1)\langle X, I\rangle$$

where the last inequality follows from the fact that $\alpha(t_2) > \alpha(t_1)$ and $\langle X, I\rangle > 0$ for any $X \in \mathcal{F}(t)$ by the second lemma. Thus

$$\langle X, N_i(0) - \alpha(t_1)I\rangle > 0$$

and $X \in \text{int}\,\mathcal{F}(t_1)$. Since this holds for any $X \in \mathcal{F}(t_2)$, we have shown that $\mathcal{F}(t_1) \supseteq \mathcal{F}(t_2)$. To show $\mathcal{F}(t_1) \supset \mathcal{F}(t_2)$, we show $\exists X' \in \mathcal{F}(t_1)$ such that $X' \notin \mathcal{F}(t_2)$.

To this end, consider $X' \in \mathcal{F}(t_1)$ such that for some i

$$\langle X', N_i(t_1)\rangle = 0$$

i.e. X' lies on the boundary of $\mathcal{F}(t_1)$. Since $N_i(t_1) = N_i(0) - \alpha(t_1)I$ we have that

$$\langle X', N_i(0)\rangle = \alpha(t_1)\langle X', I\rangle.$$

Now

$$\begin{aligned}
\langle X', N_i(0) - \alpha(t_2)I\rangle &= \langle X', N_i(0)\rangle - \alpha(t_2)\langle X', I\rangle \\
&= \alpha(t_1)\langle X', I\rangle - \alpha(t_2)\langle X', I\rangle \\
&= (\alpha(t_1) - \alpha(t_2))\,\langle X', I\rangle \\
&< 0
\end{aligned}$$

thus X' cannot be an element of $\mathcal{F}(t_2)$ and we have shown part 2.

Part 3

To show part 3 we show that $\mathsf{cone}^+(I) \subseteq \bigcap_t \mathcal{F}(t)$ and $\mathsf{cone}^+(I) \supseteq \bigcap_t \mathcal{F}(t)$. First we show that $I \subseteq \bigcap_t \mathcal{F}(t)$. For each $t \in [0,1]$, $\mathcal{F}(t)$ consists of the points X such that

$$\langle X, N_i(0) - \alpha(t)I \rangle \geq 0.$$

When $X = I$ we have

$$\langle I, N_i(0) - \alpha(t)I \rangle = \langle I, N_i(0) \rangle - n\alpha(t) \geq \langle I, N_i(0) \rangle - 1 = 0.$$

Since each set $\mathcal{F}(t)$ is a cone containing I, $\mathsf{cone}^+(I) \in \mathcal{F}(t)$ for each $t \in [0,1]$. Also, because $\mathsf{cone}^+(I)$ is in each set, it is in their intersection.

Now we shall show that $\mathsf{cone}^+(I) \supseteq \bigcap_t \mathcal{F}(t)$. Suppose there is an element Y of $\bigcap_t \mathcal{F}(t)$ not in $\mathsf{cone}^+(I)$. We may decompose Y as $Y = kI + X$ with $X \perp I$, $X \neq 0$, and $k > 0$. Since $Y \in \bigcap_t \mathcal{F}(t)$ we must have:

$$\langle N_i(t), Y \rangle = \langle N_i(0) - \alpha(t)I, kI + X \rangle \geq 0$$

for each i, and $t \in [0,1]$. Now

$$\begin{aligned} \langle N_i(0) - \alpha(t)I, kI + X \rangle &= k\mathsf{Tr}(N_i(0)) - nk\,\alpha(t) + \langle N_i(0), X \rangle - \alpha(t)\langle I, X \rangle \\ &= k(1 - n\alpha(t)) + \langle N_i(0), X \rangle. \end{aligned}$$

We can show that the second term is negative for some i. Since I is in the positive cone of the $\{N_i(0)\}$ we may write $I = \sum c_i N_i(0)$ with $c_i > 0$. Hence

$$0 = \langle I, X \rangle = \sum c_i \langle N_i(0), X \rangle$$

Since the $c_i > 0$, there must be at least one $N_i(0) = N_{i_0}(0)$ such that $\langle N_{i_0}(0), X \rangle < 0$. From above then

$$\langle N_{i_0}(t), Y \rangle \,|_{t=1} = k(1 - n\alpha(1)) + \langle N_{i_0}(0), X \rangle = \langle N_{i_0}(0), X \rangle < 0.$$

This is a contradiction. Thus there can be no $Y \in \bigcap_t \mathcal{F}(t)$ with $Y \notin \mathsf{cone}^+(I)$ so $\mathsf{cone}^+(I) \supseteq \bigcap_t \mathcal{F}(t)$. Together we have that $\mathsf{cone}^+(I) = \bigcap_t \mathcal{F}(t)$.

Part 4 Part 4 follows from parts 2 and 3.

Chapter 3

Symmetric Matrices: Reconstruction

3.1 Introduction

In this chapter we examine the inverse problem of reconstructing a symmetric matrix X given one or more projection mapping observations of the form $Y = A^T X A$, where A and Y are assumed known. The forward part of this problem was discussed in Chapter 2. We initially investigate unconstrained reconstructions, with no conditions imposed on the estimated symmetric matrix X. First some results are provided on the single observation case. We then proceed to give necessary and sufficient conditions for solution of the general case together with a complete characterization of these solutions. We rely extensively on the isometric isomorphism developed in the previous chapter in this endevor.

To be physically meaningful many problems involving symmetric matrices require these matrices to be positive semi-definite. The ellipsoid and curvature problems mentioned above are two of these. For this reason, reconstructions within the class of *positive semi-definite* (PSD) matrices are treated next. The geometry of the PSD cone examined in Chapter 2, and in particular, the finite approximations to it developed there, are used to yield *linearly* constrainted minimization problems and straightforward geometrically motivated estimation algorithms. We close the chapter by showing how to extend the PSD constrained reconstructions to the case of arbitrary interval matrix constraints.

3.2 General Symmetric Matrix Reconstruction

Chapter 2 introduced the forward problem of relating an observation Y to a matrix X through observations of the form $Y = A^T X A$. Now we consider the inverse problem, of determining X (or a set of X), given knowledge of Y and A. In our physical problems this problem often corresponds to making some (projection) observation of an object and having knowledge of the underlying observation geometry, respectively. This combination of assumptions defines a class of inverse problems of interest to us.

We begin with the case of a single observation and give necessary and sufficient conditions for the existence of a solution to the equation (2.3). Assuming such solutions exist, the set of X solving (2.3) is also characterized in a number of concise ways. Next the multiple observation case is considered, where we wish to determine X given a number of observations of the form (2.3). Again necessary and sufficient conditions are given for solution existence and a concise characterization of the solution set is provided. We conclude by considering some issues arising in the more realistic situation of solving for X given an *inconsistent* set of observations. Some recursive approaches to this task are presented at the end.

3.2.1 Single Observation

The case of a single observation of the form (2.3) is considered here. We give necessary and sufficient conditions for a solution to exist and characterize the set of solutions. We do this from a number of different perspectives. These solutions serve to motivate our use of the symmetric vector space approach introduced above for the solution of such problems. To fix notation, let $\mathcal{R}a(A) = \{y \,|\, y = Ax, x \in \text{ domain of A}\}$ denote the range of the matrix or function argument and $\mathcal{N}u(A) = \{x \,|\, 0 = Ax\}$ denote the kernel or nullspace.

Kronecker Form

Certainly one way to characterize the solutions of (2.3) is through the use of the Kronecker form equation (2.4). It is clear that a solution to (2.4) will exist if and only if $\mathsf{vec}(Y) \in \mathcal{R}a(A^T \otimes A^T)$ (the column space of $A^T \otimes A^T$). If this is satisfied, the set of vectors x satisfying $\mathsf{vec}(Y) = (A^T \otimes A^T)x$ is of the form:

$$x = (A^T \otimes A^T)^+ \,\mathsf{vec}(Y) + x_{\mathcal{N}u} \tag{3.1}$$

where $(\cdot)^+$ denotes the (unique) Moore-Penrose inverse of the argument and $x_{\mathcal{N}u}$ is any vector in $\mathcal{N}u(A^T \otimes A^T)$. When $m \leq n$ and A is full rank (a common situation) we are assured that $\mathsf{vec}(Y) \in \mathcal{R}a(A^T \otimes A^T)$ for any Y [37] and at least one solution will exist. Not all of the solutions χ in (3.1) will have the proper form of $\mathsf{vec}(X)$ imposed by the symmetry condition on X, so we must reduce the solution set above to contain only these. It is now far from clear just what this remaining set consists of.

The solution taken by Magnus and others [32, 33, 37] is to use the elimination and dilation matrices as described above to reduce the size of the problem and eliminate redundancies. Conditions may then be phrased in terms of the resulting *reduced* vector $L\,\mathsf{vec}(Y)$ and matrix $L(A^T \otimes A^T)D$, where L and D respectively are the elimination and dilation matrices described above. The set of solutions under the above conditions are then given by:

$$L\,\mathsf{vec}(X) = \left[L(A^T \otimes A^T)D\right]^+ L\,\mathsf{vec}(Y) \tag{3.2}$$

where again $(\cdot)^+$ denotes Moore-Penrose inverse. Details on the operators L, D, and the approach itself may be found in [33]. While computationally tractable, the connection between solvability conditions on these transformed matrices and the implied conditions on the original matrices is not very transparent.

Symmetric Vector Space

The above difficulties lead us to consider solutions to the problem in the vector space S_n of symmetric matrices. The pertinent equation is given by (2.6). The following result follows directly from this equation:

Result 16 (Single observation solution) *Suppose y is the representation of Y in the basis $\{M_i^{(n)}\}$ and $\tilde{A}$ is the image of A under Γ. A solution to (2.3) will exist if and only if the vector $y \in \mathcal{R}a(\tilde{A})$, In this case, the solutions are of the form:*

$$x = \tilde{A}^+ y + x_{\mathcal{N}u} \tag{3.3}$$

where $\tilde{A}^+$ is the Moore-Penrose inverse of $\tilde{A}$ and $x_{\mathcal{N}u}$ is any vector in $\mathcal{N}u(\tilde{A})$.

In particular, a solution will always exist when $m \leq n$ and $\tilde{A}$ is full rank, in which case $\tilde{A}^+ = \tilde{A}^T(\tilde{A}\tilde{A}^T)^{-1}$. Since there is a one-to-one relationship between the vectors x, y and elements of the set of symmetric matrices, it is clear that no additional conditions on them are necessary. Further the structure we have imposed on the space S and its

elements allow us insights into the nature of the result. For example, if $\{M_\ell^{(n)}\}$ is a symmetric basis for S_n, then the columns of $\tilde{A}$ are the representations of the matrices $A^T M_\ell^{(n)} A$ in the basis $\{M_\ell^{(m)}\}$. Thus for a solution to exist, we must be able to write Y as a linear combination of the matrices $A^T M_\ell^{(n)} A$, for some basis $\{M_\ell^{(n)}\}$. Further, the weights of each linear combination directly yield the representation of a solution matrix X in the original space. For example, when $n = 2$ so X is 2×2, solutions to (2.3) will exist if and only if:

$$\begin{aligned} Y &= \alpha_1 A^T \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} A + \alpha_2 A^T \begin{bmatrix} 0 & 1/\sqrt{2} \\ 1/\sqrt{2} & 0 \end{bmatrix} A + \alpha_3 A^T \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} A \\ &= A^T \left(\alpha_1 \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 & 1/\sqrt{2} \\ 1/\sqrt{2} & 0 \end{bmatrix} + \alpha_3 \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \right) A \end{aligned}$$

for some values of $[\alpha_1, \alpha_2, \alpha_3]$. Conversely, the set of all such triples represents a 2×2 matrix solving the equation.

Further, we can see that the dimension of the space of solutions is given by the dimension of the null space of the matrix $\tilde{A}$. Since the columns of $\tilde{A}$ are as given above, it is straightforward to show that if A is full rank with $m \leq n$ this dimension is equal to $n(n+1)/2 - m(m+1)/2$.

Singular Value Decomposition

We examine one final way that to characterize the solution of (2.3), through the use of the singular value decomposition (SVD) of A^T. This type of approach, utilizing the *generalized* singular value decomposition (GSVD), has recently been used to study the solutions of the equation $BXA = C$ where only X is symmetric [36]. For our case where $B = A^T$ we specialize the approach by using the SVD instead of the more general GSVD. To this end suppose $m \leq n$ and the SVD of A^T is given by:

$$A^T = U \left[\begin{array}{c|c} \Sigma & \\ \hline 0 & \end{array} \right] V^T$$

where Σ is the $r \times r$ diagonal matrix containing the r non-zero singular values of A, $r = \mathsf{rank}(A)$. Let

$$V^T X V = \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{12}^T & X_{22} & X_{23} \\ X_{13}^T & X_{23}^T & X_{33} \end{bmatrix} \begin{matrix} r \\ m-r \\ n-m \end{matrix}$$

$$U^T Y U = \begin{matrix} r & m-r & n-m \end{matrix} \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix}$$

then we have the following result:

Result 17 *Equation (2.3) with $m \leq n$ has a (symmetric) solution if and only if*

$$Y_{12} = 0 \quad \textit{and} \quad Y_{22} = 0.$$

In that case it has the general solution:

$$X = V \begin{bmatrix} \Sigma^{-1} Y_{11} \Sigma^{-1} & X_A \\ X_A^T & X_B \end{bmatrix} V^T$$

where X_A is an arbitrary $r \times (n-r)$ matrix and X_B is an arbitrary $(n-r) \times (n-r)$ symmetric matrix.

The proof of this result is in Appendix 3-A. Note in particular that the dimension of the solution space is $(n-r)(n+r+1)/2$ and that when A is full rank so that $r = m$ this reduces to $n(n+1)/2 - m(m+1)/2$, just the difference between the number of unknowns and the number of equations as stated earlier.

Consider some special cases. First, when A is an n-vector (so $m = 1$), the solution space is of dimension $n(n+1)/2 - 1$. Since the co-dimension is one, this defines a hyper-surface in the space S, specifically a hyper-*plane*. At the other extreme, when A is $n \times n$ (still full rank) so $m = n$, the solution space is of dimension zero, a point. Thus if a solution exists it must be unique. Finally, if $m = n - 1$, so A is "slightly" smaller than X, then the dimension of the solution space will be n, the size of X.

3.2.2 Multiple Observations

We now consider the case of primary interest to us: determining a symmetric matrix X from a series of observations Y_i of the form (2.3). First we examine this problem directly in the original space and then we provide a simple solution in the space of symmetric matrices that forms the basis of our results in sections to follow. We seek to solve the following problem:

Problem 1 (Symmetric matrix recovery) *Determine an $n \times n$ symmetric matrix X given A_i and Y_i such that:*

$$Y_i = A_i^T X A_i \qquad 1 \leq i \leq q \tag{3.4}$$

where each Y_i is an $m_i \times m_i$ symmetric matrix and each A_i is an $n_i \times m_i$ general matrix.

Kronecker Approach

Again we may use the Kronecker approach to understanding Problem 1, by stacking up the observations $\mathsf{vec}(Y_i)$ on the left and pulling out $\mathsf{vec}(X)$ on the right. This leads to the equation:

$$\begin{bmatrix} \mathsf{vec}(Y_1) \\ \mathsf{vec}(Y_2) \\ \vdots \\ \mathsf{vec}(Y_q) \end{bmatrix} = \left[\begin{array}{c} A_1^T \otimes A_1^T \\ \hline A_2^T \otimes A_2^T \\ \hline \vdots \\ \hline A_q^T \otimes A_q^T \end{array}\right] \mathsf{vec}(X) = \bar{A}\,\mathsf{vec}(X)$$

Conditions for uniqueness and existence of solutions may now be phrased in terms of the $(qm_i^2) \times (n^2)$ matrix $\bar{A}$ above. This approach, or variants of it using elimination/dilation matrices, can be used, but is not transparent or convenient, due to the special symmetric structure of concern to us.

Symmetric Vector Space

Instead of the above, we work in the vector space S_n of $n \times n$ symmetric matrices. In this space we may write each equation $Y_i = A_i^T X A_i$ as:

$$y_i = \tilde{A}_i x \qquad 1 \leq i \leq q$$

where y_i is the representation of Y_i with respect to a basis $\{M_\ell^{(m)}\}$, x is the representation of X with respect to a basis $\{M_\ell^{(n)}\}$, and $\tilde{A}_i$ is again the image of A_i under the map Γ. We may now stack up the observations to yield the overall equation:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_q \end{pmatrix} = \left[\begin{array}{c} \tilde{A}_1 \\ \hline \tilde{A}_2 \\ \hline \vdots \\ \hline \tilde{A}_q \end{array}\right] x$$

or

$$\mathbf{y} = \mathbf{A}x \tag{3.5}$$

where $\mathbf{y}$, $\mathbf{A}$, and x are defined in the natural way. The following characterization of the solutions of Problem 1 is now straightforward:

Result 18 (Symmetric Recovery Condition) *Problem 1 has a solution if and only if* $\mathbf{y} \in \mathcal{R}a(\mathbf{A})$. *In this case, the solutions are of the form:*

$$x = \mathbf{A}^+\mathbf{y} + x_{\mathcal{N}u}$$

where $\mathbf{A}^+$ *is the Moore-Penrose inverse of* $\mathbf{A}$ *and* $x_{\mathcal{N}u}$ *is any vector in* $\mathcal{N}u(\tilde{\mathbf{A}})$. *Further, a unique solution exists if and only if* $\mathsf{rank}\,(\mathbf{A}) = n(n+1)/2$, *in which case the solution is just* $x = \mathbf{A}^+\mathbf{y} = (\tilde{\mathbf{A}}^T\mathbf{A})^{-1}\mathbf{A}^T$ *(and* $\mathcal{N}u(\tilde{\mathbf{A}}) = \emptyset$*).*

Symmetric LLSE Problem In practice we encounter the more common case where Problem 1 has no solution at all because the observations Y_i are inconsistent. For practical reasons this is the situation of primary concern to us, reflecting the use of *noisy* measurements. When this is the case we abandon the search for exact solutions and instead seek a best approximate solution to the set (3.4). This yields the following linear least squares error problem:

Problem 2 (LLSE matrix reconstruction) *Determine the* $n \times n$ *symmetric matrix* X_{LLSE} *of minimum norm that solves the following linear least squares error (LLSE) problem:*

$$\min_X \sum_{i=1}^{q} \left\| Y_i - A_i^T X A_i \right\|_F^2 \tag{3.6}$$

where each Y_i *is an* $m_i \times m_i$ *symmetric matrix and each* A_i *is an* $n_i \times m_i$ *real matrix.*

Since $\|Y_i - A_i^T X A_i\|_F = \|y_i - \tilde{A}_i x\|_2$ for an orthonormal symmetric basis, finding the X that produces the least F-norm error is equivalent to finding the $x = x_{\text{LLSE}}$ that minimizes the following 2-norm error:

$$\|\mathbf{y} - \mathbf{A}x\|_2^2. \tag{3.7}$$

The value of the minimum is the same. Note that $\|\cdot\|_F$ is the Frobenius norm and $\|\cdot\|_2$ is the usual Euclidean norm. The solution of the transformed problem is well known:

Result 19 (LLSE Solution) *The LLSE solution x_{LLSE} of Problem 2 is given by*

$$x_{\text{LLSE}} = \mathbf{A}^+\mathbf{y}$$

where $\mathbf{A}$ and $\mathbf{y}$ are defined above and $\mathbf{A}^+$ denotes the pseudo-inverse of $\mathbf{A}$.

In the typical case where $\mathbf{A}$ has full column rank, this reduces to:

$$x_{\text{LLSE}} = \left(\sum_{i=1}^{q} \tilde{A}_i^T \tilde{A}_i\right)^{-1} \left(\sum_{i=1}^{q} \tilde{A}_i y_i\right)$$

which is the unique solution. Otherwise, the expression $\mathbf{A}^+\mathbf{y}$ yields the minimizing solution of least norm. Note that in the case of consistent observations, the minimum above is identically zero.

In terms of the original matrix space we show in Appendix 3-B that the solutions X_{LLSE} of Problem 2 must satisfy the following normal equations:

$$\sum_{i=1}^{q} A_i A_i^T X_{\text{LLSE}} A_i A_i^T = \sum_{i=1}^{q} A_i Y_i A_i^T$$

Sometimes this characterization of the set of solutions can be useful.

Again, the solution provided by Result 19 is applicable to any set of matrix observations that is linear in the matrix argument $Y_i = \mathcal{L}_i(X)$. In this case entry jk of the matrix $\tilde{A}_i$ is given by $(\tilde{A}_i)_{jk} = \langle M_j^{(m)}, \mathcal{L}_i(M_k^{(n)}) \rangle_m$, where $\{M_j\}_i$ and $\{M_k\}$ are bases for the i-th range and the domain respectively of the i-th operator. The vectors y_i and x are then the representations of the i-th observation and the unknown matrix argument, respectively.

3.2.3 Projection Parameter Space Reconstruction

We now consider an interesting method for reconstruction which involves working directly with a convenient parameterization of the projections. The underlying idea is as follows. For a fixed X, our projection mapping observations Y_i are (noisy) samples of a continuous function $Y = A^T X A$ of the mapping matrices A. Each fixed X defines a different such function of the matrix A, corresponding to a valid set of potential (noise-free) observations. Rather than trying to directly identify the symmetric matrix X from samples of one of these functions, we may instead use the samples to first reconstruct a valid observation function Y. By proper choice of a parameterization for the matrices A this step can be greatly simplified. The resulting function of A is then used to identify the underlying matrix X. This approach is reminiscent of the method of projection onto convex sets [56, 57, 58] and the general class of algorithms using iteration between spaces [59, 8], which simultaneously estimate any missing projections along with object parameters. These methods work in both the object space and the projection space. The approach is examined here because of its relationship and importance to physical problems involving support function observations. Such themes are developed further in Chapter 5.

We have completely developed the approach only for the case when $m = 1$ and $n = 2$, corresponding to scalar observations and a 2×2 X. Let the 2-vector A be a unit vector, so the corresponding projection is orthogonal. This case corresponds to reconstructing an ellipse from observations of its (squared) support function. Since A is a unit vector we may parameterize it by the angle θ so that $A = [\cos(\theta)\ \sin(\theta)]^T$. For a given X the resulting continuous observation function $Y(\theta)$ then becomes:

$$Y(\theta) = \left(\frac{X_{11} + X_{22}}{2}\right) + \sqrt{\left(\frac{X_{11} - X_{22}}{2}\right)^2 + X_{12}^2}\ \cos\left(2\theta - \tan^{-1}\left(\frac{2X_{12}}{X_{11} - X_{22}}\right)\right)$$

where X_{ij} are the entries of X and the dependence of the estimate on θ is made explicit. This expression is of the form:

$$Y(\theta) = \beta_1 + \beta_2 \cos(2\theta + \beta_3) \tag{3.8}$$

where

$$\beta_1 = \frac{X_{11} + X_{22}}{2}$$

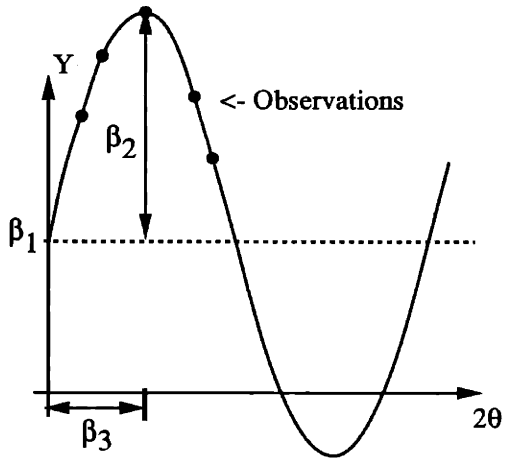

Figure 3-1: Projection parameter space situation.

$$\begin{aligned} \beta_2 &= \sqrt{\left(\frac{X_{11}-X_{22}}{2}\right)^2 + X_{12}^2} \\ \beta_3 &= -\tan^{-1}\left(\frac{2X_{12}}{X_{11}-X_{22}}\right) \end{aligned} \tag{3.9}$$

In the projection space, the function $Y(\theta)$ traces out an offset phase-shifted cosine, as illustrated in Figure 3-1. In the situation of Problem 2 we would observe a finite number of noise corrupted points $Y_i(\theta_i)$ of this curve, as shown. The fact that

$$\min_{X=X^T} \sum_{i=1}^{q} \|Y_i - A_i^T X A_i\|_F = \min \sum_{i=1}^{q} |Y_i - Y(\theta)|$$

allows us to proceed by carrying out the minimization on the right. Rather than directly reconstructing the matrix X, as in the minimization on the left, we first reconstruct the best fitting function $\hat{Y}(\theta)$ in the projection space, then extract the elements of the matrix X_{ij} from the parameters of this function using the relationships:

$$\begin{aligned} X_{11} &= \beta_1 + \beta_2 \cos(\beta_3) \\ X_{22} &= \beta_1 - \beta_2 \cos(\beta_3) \\ X_{12} &= -\beta_2 \sin(\beta_3) \end{aligned} \tag{3.10}$$

Note that we may also express (3.8) as a *linear* function of a different set of variables as follows:

$$Y(\theta) = \tilde{\beta}_1 + \tilde{\beta}_2 \cos(2\theta) + \tilde{\beta}_3 \sin(2\theta)$$

which is linear in the parameters $\tilde{\beta}_1$, $\tilde{\beta}_2$, and $\tilde{\beta}_3$. The problem is then another LLSE problem. The formulation of (3.8) is actually easier to work with when we introduce positivity constraints into the problem.

For the $n = 3$ case (3×3 matrices) with 1-dimensional observations, the unit vector A requires two parameters to characterize, since $A(\theta, \phi)$ is actually a parameterization of the unit sphere. For the standard latitude and longitude parameterization of the unit sphere, the resulting function $Y(\theta, \phi)$ becomes a harmonic surface of order one in one of the variables and order two in the other variable. Unfortunately, this function is not separable in a nice way in these variables. The appropriate expressions are provided in Appendix 3-C. We have not used these higher order expressions to date.

3.2.4 Recursive Approaches

We may also use the formulation of (3.7) to implement a recursive solution to Problem 2. Because of the structure of (3.7) such a solution is just a recursive least squares formulation with the state vector being the estimate $\hat{x}$ of the symmetric matrix represented by x (for an appropriate choice of initial condition). These equations are given by:

$$\begin{aligned} \hat{x}_k &= \hat{x}_{k-1} + K_k \left(y_k - \tilde{A}_k \hat{x}_{k-1} \right) \\ K_k &= Q_k^{-1} \tilde{A}_k^T R_k \\ Q_k^{-1} &= R_{k-1}^{-1} - R_{k-1}^{-1} \tilde{A}_k^T \left(R_k^{-1} + \tilde{A}_k R_{k-1}^{-1} \tilde{A}_k^T \right)^{-1} \tilde{A}_k R_{k-1}^{-1} \end{aligned} \tag{3.11}$$

where $\hat{x}_k$ is the representation of the LLSE estimate of the matrix X using observations up to time k, y_k is the representation of the observation Y_k, $\tilde{A}_k$ is the image of the matrix A_k under Γ, and the R_k are optional weighting matrices reflecting our confidence in observation k. The solution to (3.11) minimizes the quantity:

$$\hat{x}_0^T Q_0 \hat{x}_0 + \sum_{i=1}^{q} (y_k - \tilde{A}_k \hat{x}_{k-1})^T R_i (y_k - \tilde{A}_k \hat{x}_{k-1})$$

For the unweighted LLSE case $R_k = I$. Stochastic interpretations result if the weighting matrices R_k are chosen to be observation error covariance matrices [60].

This recursive formulation, while introduced for the case of estimating a static matrix X, suggests another interpretation. Since the estimate changes from step to step, we have created a *dynamic* symmetric matrix in the estimates $\hat{X}_i$. Further, we can imagine that rather than the target matrix X being static as above, it evolves according to some dynamical equation. Such a situation models the evolution of an ellipsoid in Chapter 5. We defer further discussion until that chapter.

3.3 PSD Reconstruction

As discussed in Section 2.3, many physical problems fitting into the symmetric matrix reconstruction framework that we have been considering require that the underlying matrix be positive semi-definite. This requirement is necessary, for example, in the case of curvature representation of convex surfaces (where the semi-definiteness reflects the convexity of the surface) and for ellipsoid representation (where semi-definiteness of the underlying matrix ensures the object is an ellipsoid). We study both problems in more detail later. Such considerations lead us to study the reconstruction of *positive semi-definite* (PSD) symmetric matrices in this section. We use the insights that we have developed into the PSD cone and our approximations of it to suggest ways to solve least-squares problems subject to a PSD constraint on the solution. Specifically, we are concerned with the following variant of Problem 2:

Problem 3 (PSD matrix reconstruction) *Determine the $n \times n$ symmetric PSD matrix X_{PSD} of minimum norm that solves the following linear least squares error (LLSE) problem:*

$$\min_{X=X^T \geq 0} \sum_{i=1}^{q} \left\| Y_i - A_i^T X A_i \right\|_F^2 \tag{3.12}$$

where each Y_i is an $m_i \times m_i$ symmetric matrix and each A_i is an $n_i \times m_i$ real matrix.

First we show that our work on the *unconstrained* least-squares problem presented in Section 3.2 is useful as it stands for many practical cases. If our observations (the projection mappings Y_i) arose from an underlying physical process involving a PSD matrix, then with noise-free observations, this is exactly the matrix the methods discussed previously would recover. We show that when the deviations from this assumption are small, in a sense to be made precise, the solution will be close to the original unperturbed PSD solution. Thus, for many problems no alterations are needed to the unconstrained approaches. This will not always be the case however,

and then it becomes necessary to incorporate the PSD constraint directly into the reconstruction process in some way.

For this situation, we use the insights into and approximations of the PSD cone that we have developed to yield straightforward constrained optimization problems. We use polyhedral approximations to the PSD set, which in turn lead to linear constraints on the elements of a matrix. These linear constraints may then be incorporated into the least squares formulation developed above to yield straightforward constrained optimization problems. Finally we use our insight into the PSD cone to present some iterative and gradient based approaches to this problem that exploit the structure of the problem to improve on the approximate solutions.

3.3.1 Small Noise Case

The solution developed in Result 19 has no constraint to guarantee the positive semi-definiteness of the solution X_{LLSE}. Still, in many instances, the observations are "good enough" that the underlying positive semi-definite nature of the solution is maintained. In particular, if we hypothesize an underlying matrix which is positive semi-definite (from physical considerations perhaps), and if our observations in (2.3) were perfect, we would recover precisely the original PSD matrix. In the more realistic case that the observations are noisy and inconsistent, we may think of this noise as a perturbation to the above perfect situation. From continuity considerations, we would expect that if this perturbation or noise in the ideal set of observations were small enough, the solution obtained by Result 19 would still be positive semi-definite. We make these notions precise in the following.

Result 20 (Sensitivity bounds) *Suppose that the symmetric matrices X^* and $X^*+\delta X$ are the unique solutions to:*

$$\begin{aligned} X^* &= \arg\min \sum_{i=1}^{q} \left\| Y_i^* - A_i^T X A_i \right\|_F^2 \\ X^* + \delta X &= \arg\min \sum_{i=1}^{q} \left\| (Y_i^* + \delta Y_i) - A_i^T X A_i \right\|_F^2 \end{aligned}$$

where A_i is $n \times m$ and full rank. Then:

$$\sigma_{\max}(\delta X) \leq \sigma_{\min}^{-1}(\mathbf{A}) \sqrt{\sum_{i=1}^{q} \|\delta Y_i\|_F^2}$$

where $\mathbf{A} = [\tilde{A}_1^T|\tilde{A}_2^T|\ldots|\tilde{A}_q^T]^T$, $\tilde{A}_i = \Gamma(A_i)$ *(see (3.5)).*

Further, if X^* *is positive definite and*

$$\sigma_{\min}^{-1}(\mathbf{A})\sqrt{\sum_{i=1}^{q}\|\delta Y_i\|_F^2} \leq \lambda_{\min}(X^*)$$

then $X^*+\delta X$*, the LLSE solution of the perturbed problem, will also be positive definite. Here* $\lambda_{\min}(\cdot)$ *denotes the minimum eigenvalue of its argument and* $\sigma_{\min}(\cdot)$ *and* $\sigma_{\max}(\cdot)$ *denotes the minimum and maximum singular value of the argument, respectively.*

The proof of this result is in Appendix 3-D. The result says that if our observation geometry is good, as reflected in a large value of $\sigma_{\min}(\mathbf{A})$, if the noise is small, as reflected in small values of $\|\delta Y_i\|_F$, and if the underlying true matrix X^* is not close to singular, so $\lambda_{\min}(X^*)$ is large, then the solution provided by Result 19 will still yield a PSD matrix. Small values of $\sigma_{\min}(\mathbf{A})$ can reflect near linear dependence of observations, large values of $\|\delta Y_i\|_F$ result from noisy observations, and small values of $\lambda_{\min}(X^*)$ can reflect extremes of the underlying physical objects. For an ellipsoid, a small value of $\lambda_{\min}(X^*)$ might reflect an ellipse which is almost degenerate, while for the case of surface curvature representation, such small values can occur at points of high curvature.

In spite of the above arguments, there may be situations when, due to noisy observations, the LLSE solution is not positive semi-definite. Such a situation can arise even when all our observations themselves are positive semi-definite, as the following example shows. If $A_1 = [0\ \ 1]^T$, $A_2 = \frac{1}{\sqrt{10}}[3\ \ 1]^T$, $A_3 = \frac{1}{\sqrt{2}}[1\ \ -1]^T$, and $Y_1 = 10$, $Y_2 = .1$, $Y_3 = 4.5$ then clearly the observations are all positive, yet $X_{\text{LLSE}} = \mathbf{diag}[-1\ \ 10]$, is an indefinite diagonal matrix. This possibility is easy to see in the planar case. As discussed in Section 3.2.3, we may view this case as trying to find the best fitting cosine from noisy observations of points on the curve. These observations may certainly be positive yet the best fitting cosine may not always be, as shown in Figure 3-1. Because of such difficulties we consider constraints and procedures to ensure a positive semi-definite solution to (3.12) next.

3.3.2 Non-Iterative Techniques

In this section we consider non-iterative techniques for finding a solution to the PSD constrained least squares problem. These methods yield an approximation to the true minimizing PSD matrix which solves Problem 3.

Nearest PSD Matrix

The simplest approach we might take to approximating the solution to Problem 3, would be to find the closest matrix X_{near} in the PSD set to the solution X_{LLSE} of the unconstrained Problem 2. Algebraically, the nearest PSD matrix $[Z]^{\text{PSD}}$ to an arbitrary symmetric matrix Z is given by

$$[Z]^{\text{PSD}} = \underset{X \in \text{PSD}_n}{\arg\min} \|X - Z\|_F^2$$

and corresponds to projecting Z onto the PSD cone. If $\lambda_i(Z)$ are the eigenvalues of Z and v_i are the corresponding eigenvectors, then it is straightforward to show that

$$[Z]^{\text{PSD}} = \sum_{\lambda_i(Z) \geq 0} \lambda_i(Z)\, v_i v_i^T \tag{3.13}$$

Thus, for the present case we would obtain $X_{\text{near}} \equiv [X_{\text{LLSE}}]^{\text{PSD}}$ from X_{LLSE} by setting its negative eigenvalues to zero. While this solution is not optimal, it is easy to compute, and as such may be used to convert any estimate which is not positive semi-definite into a PSD estimate. We may also use such an estimate to obtain a starting value for other methods.

Note that, in general, $X_{\text{near}} \neq X_{\text{PSD}}$, so this solution is just an approximation to the true solution. The projection onto the PSD cone to obtain the minimizing PSD constrained solution may be decomposed into two orthogonal projections, one corresponding to the determination of the unconstrained LLSE solution, as shown in Figure 3-2. As a result we have that

$$X_{\text{PSD}} = \underset{X \in \text{PSD}_n}{\arg\min} \sum_{i=1}^{q} \left\| A_i^T (X - X_{\text{LLSE}}) A_i \right\|_F^2$$

while X_{near} is given by

$$X_{\text{near}} = \underset{X \in \text{PSD}_n}{\arg\min} \|X - X_{\text{LLSE}}\|_F^2$$

These two expressions are not the same, in general. What we want to do is to project the *image* of the LLSE estimate in the column space of the transformation ($\mathbf{A}x_{\text{LLSE}}$ in the figure) onto the *image* of the constraint set. This is not the same as projecting the LLSE estimate directly onto the constraint set. In the original space the PSD constraint is straightforward to specify but the minimization is difficult (due to the

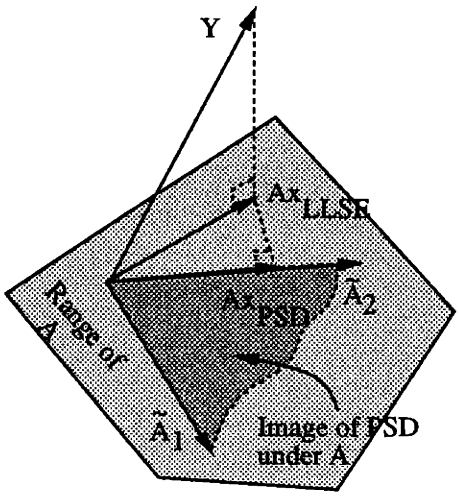

Figure 3-2: Geometry of solution determination.

presence of **A**) while in the range space of the transformation the minimization is just a projection onto the image of the PSD constraint, but this transformed constraint set is now difficult to specify.

Approximate Reconstruction

Here we consider a different approach to the constrained reconstruction problem, using the approximations of the PSD cone developed above. Instead of requiring the solution to be a PSD matrix, we impose the constraint that the solution $X_{\widetilde{\text{PSD}}}$ lie in one of the approximating sets developed in Section 2.4. In other words, we solve exactly the following perturbed problem, whose solution approximately solves Problem 3:

Problem 4 (Approximate PSD Matrix Reconstruction) *Determine the $n \times n$ symmetric matrix X of minimum norm that solves the following constrained linear least square error (LLSE) problem:*

$$\min_{X \in \widetilde{\text{PSD}}_n} \sum_{i=1}^{q} \left\| Y_i - A_i^T X A_i \right\|_F^2 \tag{3.14}$$

where $\widetilde{PSD}_n$ is a set approximating the PSD set and each Y_i is an $m_i \times m_i$ symmetric matrix and each A_i is an $n_i \times m_i$ real matrix.

The type of PSD cone approximation $\widetilde{\mathrm{PSD}}_n$ used obviously has a direct bearing on the properties of the resulting solution. For example, using a finite support approximation as found in Lemma 2 allows solutions that are not positive semi-definite. Such properties should be obvious from the nature of the approximating sets and our discussion of them.

Note that the constraints imposed by any of the static approximations $\widetilde{\mathrm{PSD}}_n$ that we developed in Section 2.4 can be separated into just two groups for optimization purposes. If $x = \Xi(X)$ is the representation of the symmetric matrix X with respect to some symmetric basis $\{M_\ell\}$, then the condition that $X \in \widetilde{\mathrm{PSD}}_n$ for any of the approximating sets developed above can be expressed in one of the two following forms:

1. Linear inequality constraint on x:

$$Ex \geq f$$

2. Linear inequality constraint on a linearly transformed variable $\bar{x}$:

$$\begin{aligned} x &= G\bar{x} \\ E\bar{x} &\geq f \end{aligned}$$

The first group yields linear inequality constraints directly in terms of the elements of the problem x while the second group yields linear inequality constraints in terms of the elements of a linearly transformed variable $\bar{x}$. The different approximating sets, their type, and the corresponding values of the variables E, f, and G are presented in Appendix 3-E. More detail on the respective approximations may be obtained from the section discussing that approximation.

If $y_i = \Xi(Y_i)$ is the representation of the symmetric matrix Y_i with respect to some symmetric basis $\{M_\ell\}$, $\mathbf{y} = [y_1^T | y_2^T | \cdots | y_q^T]^T$, and $\tilde{\mathbf{A}} = [\tilde{A}_1^T | \tilde{A}_2^T | \cdots | \tilde{A}_q^T]^T$, where $\tilde{A}_i = \Gamma(A_i)$, then Problem 4 may always be written in one of following two forms

$$\min_{Ex \geq f} \|\mathbf{y} - \mathbf{A}x\|_2^2$$

or

$$\min_{E\bar{x} \geq f} \|\mathbf{y} - \mathbf{A}G\bar{x}\|_2^2$$

for some E, f, and perhaps G depending on the underlying approximating set. Both of these forms are least squares problems with linear inequality constraints, which are readily solved. Later we present some numerical comparisons between estimates based on these methods and those of other methods.

3.3.3 Iterative Methods

We next examine methods that are iterative in nature and *not* finitely terminating. In contrast to the above approach of exactly solving a perturbed problem, these methods attempt to approximately solve the exact Problem 3. First we treat methods not directly using a gradient of the cost function. In the section after that, we examine approaches that do incorporate such gradient information.

Successive Halfspaces

In Section 2.3.1 we discussed representing the PSD cone through its supporting halfspaces. This approach yielded an exterior representation of the cone. We further showed that only the halfspaces with normals corresponding to rank one matrices were needed. Unfortunately, to fully represent the PSD cone required an infinite set of these halfspaces. As a result, we considered approximating the set by using a finite number of these supporting halfspaces in Section 2.4.2. The estimate based on any such static approximation may, in general, be indefinite. Note, however, that when $\mathbf{A}$ is of full column rank, *if* the resulting estimate based on such an external approximating set happens to be positive semi-definite, then it must also be the unique optimal solution to the PSD constrained Problem 3 (recall that $\mathbf{A} = [\tilde{A}_1^T|\tilde{A}_2^T|\cdots|\tilde{A}_q^T]^T$, with $\tilde{A}_i = \Gamma(A_i)$). This property is simply a consequence of the inclusion of the PSD set in any external approximating set. If the minimizing solution over the larger set is a PSD matrix then, clearly, restricting the solution to be a PSD matrix will not increase the cost.

The interesting case is when, for a given approximating set, the solution of Problem 4 lies near the set boundary. If the solution is near the center of the set (and is hence PSD), then from the above argument we are done anyway. As we add more support halfspaces the approximation to the PSD cone should get better, but how should we choose which support halfspaces to add? Clearly, adding support halfspaces on the side of the cone opposite the current solution is inefficient, since these constraints will not be binding. We investigate this question and develop an algorithm

for choosing halfspaces guaranteed to converge to the solution of Problem 3.

As we said, when the solution to Problem 4 is PSD we have found the optimal solution, so we consider the case of indefinite solutions. Any symmetric matrix X can be written as the following weighted sum of rank 1 matrices:

$$X = \sum_{i=1}^{r} \lambda_i v_i v_i^T$$

where r is the rank of the matrix, v_i are the eigenvectors, and λ_i are the associated eigenvalues. Now consider the constraint imposed by the support halfspace $\mathcal{HS}(v_i v_i^T, 0)$ for some i. This constraint is given by

$$\langle v_i v_i^T, X \rangle \geq 0$$

or, equivalently

$$\begin{aligned} v_i^T X v_i &= v_i^T \left(\sum_{j=1}^{r} \lambda_j v_j v_j^T \right) v_i \\ &= \lambda_i \geq 0 \end{aligned}$$

where the last equality follows from the fact that the eigenvectors of a symmetric matrix are mutually orthogonal. Thus adding the constraint corresponding to the halfspace $\mathcal{HS}(v_i v_i^T, 0)$ is the same as requiring that all matrices of the corresponding set with the eigenvector v_i have a non-negative corresponding eigenvalue!

This insight suggests the following approach. First solve Problem 4 with a given external approximating set. Then look at the eigenvalues of the corresponding solution $X(k)$ and add to the current approximating set those halfspaces with rank-1 normals corresponding to the eigenvectors of the negative roots. At the next stage then, either these roots must be non-negative or the eigenvectors must have changed. One way of viewing this idea is as follows. Using a set of support halfspaces, we form a good approximation to the PSD cone boundary locally to the solution of Problem 3. The solution to the approximate problem at any stage tells us the part of the PSD cone that is important to approximate well. This reasoning gives us the following algorithm for finding a solution to Problem 3 .

Algorithm 1 (Successive Halfspace)

Step 1) *Choose an initial external approximating set of hyperplane normals* $\{N_i\}_0$.

Set $k = 0$.

Step 2) *Solve Problem 4 with the approximating set* $\widetilde{PSD}_n(k) = \bigcap_{N \in \{N_i\}_k} \mathcal{HS}(N, 0)$ *to obtain* $X(k)$.

Step 3) *Find the negative eigenvalues* λ_j *and corresponding eigenvectors* v_j *of* $X(k)$. *If there are none, the solution is optimal: STOP.*

Step 4) *Update the approximating set* $\{N_i\}_k$ *by adding halfspaces with rank-1 normals corresponding to the negative eigenvalues:* $\{N_i\}_{k+1} = \{N_i\}_k \bigcup \{v_j v_j^T\}$ *for* $\lambda_j < 0$, *thus* $\widetilde{PSD}_n(k+1) = \widetilde{PSD}_n(k) \bigcap \mathcal{HS}(v_j v_j^T, 0)$.

Step 5) *Goto Step 2).*

We make some observations about the steps of the algorithm. For Step 1) *any* polyhedral external approximating set $\widetilde{\mathrm{PSD}}_n(k)$ defined by a group of linear inequality constraints, one for each bounding halfspace, can be used. Thus, Step 2) requires the solution of an LLSE problem subject to these linear inequality constraints. This problem, known as problem LSI [55], is straightforward to solve. For Step 3) the eigenanalysis of an $n \times n$ matrix is required, which is also straightforward to perform. Further, it is reasonable to expect that as the iteration proceeds, and the solution converges, we could simply update the previous eigenanalysis using, for example, first order perturbation expressions for the eigenvalues and eigenvectors or one step of a Rayleigh-quotient iteration [61]. Finally, Step 4) essentially reduces to appending additional rows to the constraint matrix of the linear inequalities ($Ex \geq f$). The point to note is that the constraint set (and thus this matrix) grows linearly with the iteration count.

Note that the algorithm is, in general, non-terminating. Further, the estimate at any finite stage of the algorithm $X(k)$ will, in general, be indefinite since the constraint set $\widetilde{\mathrm{PSD}}_n(k)$ will *always* form an *external* approximation to the PSD set for any finite number of steps and thus contain elements outside the PSD set. We may easily terminate the algorithm after a finite number of steps, for example, when the change between iterations falls below some threshold, then project the corresponding (indefinite) iterate $X(k)$ onto the PSD cone (by finding the nearest PSD matrix to $X(k)$ as described in (3.13) of Section 3.3.2) to obtain a final PSD estimate. The error introduced by this final projection is small if $X(k)$ is near the optimal solution (as reflected in the diminishing size of the updates).

We have the following result concerning the convergence of Algorithm 1.

Result 21 (Hyperplane Algorithm Convergence) *If* $\mathbf{A}$ *is of full column rank, then Algorithm 1 converges monotonically to the unique solution to Problem 3.*

The proof is in Appendix 3-F. Numerical examples are discussed in detail later. In practice it appears that when $n = 2$ just 20 or so iterations are sufficient to provide convergence (often far less).

Finite Successive Hyperplanes

In Algorithm 1 the constraint set grows linearly with the iteration count. It would obviously be better if this could be avoided. Heuristically, if the algorithm has proceeded for some time and the iterates $X(k)$ have settled down toward some value, it might be reasonable to expect that the constraints from the distant past are no longer necessary. Essentially, we would argue that the most recent support halfspace constraints effectively capture the salient parts of the local PSD cone geometry. Rather than keep *all* of the constraints in Algorithm 1 we would only retain the n_{const} most recent ones. As the iteration proceeds these n_{const} halfspaces should form a progressively better approximation to the local structure of the cone at the solution. Of course, with too few constraints we would not expect to retain enough of the local information either.

Currently, we have no results on the minimum value of n_{const} required for convergence of a given problem. For typical problems an acceptable value seems to be in the range of $10 \sim 20$. When this number is chosen too small, the iterations converge for a time, then, as some critical constraints are removed from the set, the iterate jumps back to something near a previous value, corresponding to lower cost (recall the cost should be *increasing*) and repeats this cycle of irregular (almost chaotic) oscillation. For certain problems the convergence phase can proceed for quite some time (to the eye even to completion) before this divergence is observed. One possibility to prevent this behavior is to stop the iteration if the residual stops decreasing monotonically or to change the number of constraints retained n_{const} to a larger number on the fly. For example, we could delete constraints as long as the residual is decreasing and retain them if the residual starts to increase.

Unconstrained Square-Root Iteration

Next we consider methods explicitly using the cost function of Problem 4 and its gradient. Recall that the cost function $C(X)$ is given by:

$$C(X) = \sum_{i=1}^{q} \left\| Y_i - A_i^T X A_i \right\|_F^2$$

Perhaps the simplest thing to do is to work with a square root of the matrix X. A square root of a symmetric positive semi-definite matrix X is any matrix R such that $X = R^T R$. Conversely, for any real matrix R, the matrix $R^T R$ is guaranteed to be positive semi-definite. Unfortunately for a given X the corresponding R is not unique. Still, we may search for any R such that the corresponding $X = R^T R$ minimizes $C(X)$. This approach yields the following equivalent form of Problem 3 in terms of the square root matrix R and the cost $C(R)$:

Problem 5 (Square Root PSD Matrix Reconstruction) *Determine an $n \times n$ real general matrix R_{OPT} that solves the following unconstrained linear least squares error (LLSE) problem:*

$$\min \sum_{i=1}^{q} \left\| Y_i - A_i^T R^T R A_i \right\|_F^2 \tag{3.15}$$

where each Y_i is an $m_i \times m_i$ symmetric matrix and each A_i is an $n_i \times m_i$ real matrix.

Note that Problem 5 is now an *unconstrained* optimization problem in terms of a general matrix and that the solution of Problem 3 is given by $X_{\mathrm{PSD}} = R_{\mathrm{OPT}}^T R_{\mathrm{OPT}}$. Thus we have transformed the constrained optimization problem into an unconstrained problem involving a general matrix. We may therefore immediately use any procedure for finding unconstrained minima. One example would be the readily available Nelder-Mead Simplex Algorithm [62], which numerically descends the cost surface. Unfortunately, this algorithm makes no use of the special structure of the problem, and is very slow. By using analytic gradient information we may considerably speed up convergence. Note that the solution square-root R_{OPT} will not be unique but the corresponding symmetric matrix $X_{\mathrm{PSD}} = R_{\mathrm{OPT}}^T R_{\mathrm{OPT}}$ will be.

Unconstrained Steepest Descent

One approach we may take is to perform an (unconstrained) steepest descent [63] on the cost function $C(R)$ to find a solution to Problem 5. To do this we require an

analytic expression for the gradient $\nabla C(R)$ of $C(R)$. This quantity is evaluated in Appendix 3-G and is given by

$$\nabla C(R) = -4R\sum_{i=1}^{q} A_i \left(Y_i - A_i^T R^T R A_i\right) A_i^T. \tag{3.16}$$

With the gradient we may define the following algorithm:

Algorithm 2 (Steepest Descent on R)

Step 1) *Choose an initial square root $R(0)$. Set $k = 0$.*

Step 2) *Evaluate $\nabla C(R(k))$.*

Step 3) *Set $R(k+1) = R(k) - \gamma \nabla C(R(k))$.*

Step 4) *Goto 2).*

One way of choosing the initial value for Step 1) is to use a square root of the PSD estimate X_{near} based on the unconstrained $LLSE$ estimate. We have left the choice of the stepsize γ in Step 3) open.

Instead of using a constant stepsize γ in Step 3) of the algorithm we could use a stepsize that leads to the largest possible reduction in the size of C at any stage [63]. We would choose $\gamma_k = \gamma$ in Step 3) to minimize the cost at the next step of the algorithm. That is, to minimize $C(R(k) - \gamma \nabla C(R(k)))$. This is a 1-dimensional minimization that must be performed at each iteration. Many authors reserve the term steepest descent for algorithms employing this choice of γ at each iteration [64, 65, 66]. For our particular problem the search for the minimizing γ_k reduces to finding a root of a cubic. The necessary calculations and coefficients of the appropriate cubic in terms of the elements of the problem are provided in Appendix 3-G. This choice of γ_k often yields the fastest convergence of Algorithm 2.

We have the following result concerning the convergence of Algorithm 2 with the above stepsize selection rule.

Result 22 (Convergence of Algorithm 2) *In Algorithm 2 if* $\mathbf{A}$ *is of full column rank and γ_k is chosen to minimize $C(R(k) - \gamma \nabla C(R(k)))$ at each iteration*

$$\lim_{k \to \infty} \nabla C\left(R(k)\right) = 0$$

and the algorithm is guaranteed to converge.

The proof is in Appendix 3-G. Unfortunately, the cost function $C(R)$ is not convex in R and thus while we know that Algorithm 2 will converge, we cannot guarantee that the limit point is a global minimum of $C(R)$. Next we investigate a constrained gradient method which *does* have such optimality guarantees. Note that for a general square root R there are also almost twice as many variables to find vs finding the entries of a symmetric matrix.

Constrained Gradient Projection

We now examine a different approach. Rather than searching for an unconstrained square root we directly search for a solution to the constrained problem using the cost $C(X)$. The simplest approach we might take is to use the gradient projection algorithm [63]. This algorithm is much like the steepest descent algorithm. We make an update along the gradient direction, but since an update of this type could take us outside the constraint set, we take the additional step of projecting back into the set at every iteration. This projection operation simply corresponds to finding the nearest PSD matrix to the current estimate and is given by equation (3.13). Recall that this projection is done by setting the negative eigenvalues to zero and is denoted by $[\cdot]^{\text{PSD}}$.

As before we need the gradient $\nabla C(X)$, this time directly with respect to X. This expression is evaluated in Appendix 3-G and is given by:

$$\nabla C(X) = -2\sum_{i=1}^{q} A_i \left(Y_i - A_i^T X A_i\right) A_i^T. \tag{3.17}$$

With this gradient we may define the following algorithm:

Algorithm 3 (Gradient Projection on X)

Step 1) *Choose an initial guess* $X(0)$. *Set* $k = 0$.

Step 2) *Evaluate* $\nabla C(X(k))$.

Step 3) *Take a step:* $W(k) = X(k) - \gamma \nabla C(X(k))$.

Step 4) *Project back onto the PSD set to obtain the next iterate:* $X(k+1) = [W(k)]^{\text{PSD}}$.

Step 5) *Goto 2).*

We may once again choose the initial value $X(0)$ to be the PSD estimate X_{near} based on the unconstrained $LLSE$ estimate. Again, there is the issue of the choice of stepsize γ in Step 3). The next convergence result provides a bound on its value.

We have the following result concerning the convergence of Algorithm 2.

Result 23 (Convergence of Algorithm 3) *In Algorithm 3 if* $\mathbf{A}$ *has full column rank and* $0 < \gamma < 2/\alpha$, *where*

$$\alpha = \sigma^2_{\max}(\mathbf{A})$$

and if X^* *is a limit point of the sequence* $\{X(k)\}$ *generated by the algorithm, then* X^* *is the unique minimizer of* $C(X)$ *over the PSD cone. Further if* γ *is chosen small enough, the sequence* $\{X(k)\}$ *converges to* X^* *geometrically.*

Here $\mathbf{A}$ is as defined previously and $\sigma_{\max}(\mathbf{A})$ denotes the maximum singular value of $\mathbf{A}$. The proof is in Appendix 3-G.

Again, an alternative to using a constant stepsize in Step 3) is to use, at each iteration, the stepsize that leads to the largest possible reduction in the cost C. That is, at each iteration we would choose γ to minimize $C(X(k) - \gamma \nabla C(X(k)))$. For the present problem this 1-dimensional minimization may be carried out analytically to yield for the optimal $\gamma_{\min}$:

$$\gamma_{\min} = \frac{1}{2} \frac{\|\nabla C(k)\|_F^2}{\sum_{i=1}^q \|A_i^T \nabla C(k) A_i\|_F^2}.$$

This quantity appears difficult to compute but since $\nabla C(k)$ is now a symmetric matrix we may find an equivalent expression in the symmetric vector space which is perhaps more transparent. If we define $\nabla c(k) = \Xi(\nabla C(k))$ as the representation of this symmetric matrix, and define $\mathbf{A}$ as in (3.5), then $\gamma_{\min}$ is simply given by:

$$\gamma_{\min} = \frac{1}{2} \frac{\|\nabla c(k)\|_2^2}{\|\mathbf{A} \nabla c(k)\|_2^2}.$$

As mentioned above, this choice of γ_k often yields the fastest convergence of Algorithm 3, at the expense of the additional computation of $\gamma_{\min}$.

Projection Parameter Space Approach

Here we examine a specialized algorithm for the $n = 2$ and $m = 1$ case of 2×2 matrices and 1-dimensional observations. This algorithm utilizes the projection space approach

discussed in Section 3.2.3. Recall that for this case when $A_i(\theta_i) = [\cos(\theta_i) \;\; \sin(\theta_i)]^T$ our scalar observations become

$$Y_i(\theta_i) = \beta_1 + \beta_2 \cos(2\theta_i + \beta_3)$$

where

$$\beta_1 = \frac{X_{11} + X_{22}}{2} \tag{3.18}$$

$$\beta_2 = \sqrt{\left(\frac{X_{11} - X_{22}}{2}\right)^2 + X_{12}^2} \tag{3.19}$$

$$\beta_3 = -\tan^{-1}\left(\frac{2X_{12}}{X_{11} - X_{22}}\right). \tag{3.20}$$

See Figure 3-1. Originally, we were not concerned with imposing a positive semi-definite constraint on the solution. By definition, X is positive semi-definite if and only if the reconstructed function $\hat{Y}(\theta) \geq 0$ for all θ. Recovering a (2×2) positive semi-definite X from 1-D observations of $Y(\theta_i)$ is thus equivalent to solving the following projection space problem:

$$\min_{\beta_1 \geq |\beta_2|} \sum_{i=1}^{q} |Y_i - [\beta_1 + \beta_2 \cos(2\theta_i + \beta_3)]|^2 \tag{3.21}$$

where we are given the Y_i and θ_i and must find the parameters β_1, β_2, β_3. In the projection space then, we wish to reconstruct a nonnegative cosine given noisy observations of points along the the cosine. As given above, the problem has linear constraints but nonlinear cost. We may also write this problem as one with nonlinear constraints and a linear cost, but the present form is actually easier to work with.

If we knew the value of β_3, solving (3.21) for the minimizing value of β_1 and β_2 would be easy. Conversely, given β_1 and β_2 we can easily find corresponding minimizing β_3 since

$$\beta_1 + \beta_2 \cos(2\theta_i + \beta_3) = \beta_1 + \beta_2 \cos(2\theta_i) \;\cos(\beta_3) - \beta_2 \sin(2\theta_i) \;\sin(\beta_3).$$

Each of these steps is a constrained least-squares problem. This alternation forms the basis of our projection space PSD reconstruction algorithm for the $n = 2$ case.

Algorithm 4 (PSD Projection Space)

Step 1) *Choose an initial guess $X(0)$ and find $\beta_3(0)$ from (3.20). Set $k = 0$.*

Step 2) *Solve the following linearly constrained LLSE problem for* $\beta_1(k+1)$, $\beta_2(k+1)$*:*

$$\begin{bmatrix} \beta_1(k+1) \\ \beta_2(k+1) \end{bmatrix} = \underset{\beta_1 \geq \beta_2 \geq 0}{\arg\min} \left\| \mathbf{Y} - \mathbf{M}(k) \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \right\|_2^2$$

where $\mathbf{Y} = [Y_1|Y_2|\cdots|Y_q]^T$*, and the matrix* $\mathbf{M}(k)$ *is given by*

$$\mathbf{M}(k) = \begin{bmatrix} 1 & \cos(2\theta_1 + \beta_3(k))] \\ 1 & \cos(2\theta_2 + \beta_3(k))] \\ \vdots & \vdots \\ 1 & \cos(2\theta_q + \beta_3(k))] \end{bmatrix}.$$

Step 3) *Solve the following quadratically constrained LLSE problem for* z^**:*

$$z^* = \underset{\|z\|=1}{\arg\min} \left\| \tilde{\mathbf{Y}} - \mathbf{N}(k)z \right\|_2^2$$

where $\tilde{\mathbf{Y}} = [\mathbf{Y} - \beta_1(k+1)]/\beta_2(k+1)$*, and the matrix* $\mathbf{N}(k)$ *is given by*

$$\mathbf{N}(k) = \begin{bmatrix} \cos(2\theta_1) & -\sin(2\theta_1)] \\ \cos(2\theta_2) & -\sin(2\theta_2)] \\ \vdots & \vdots \\ \cos(2\theta_q) & -\sin(2\theta_q)] \end{bmatrix}.$$

Step 4) *Solve for* $\beta_3(k+1) = \angle z = \arctan(z_2/z_1)$.

Step 5) *Form* $X(k+1)$ *using (3.10).*

Step 6) *Goto Step 2).*

This can be seen to be a coordinate descent (nonlinear Gauss-Seidel) algorithm [63] for fitting a positive cosine to data. In step 2) we hold the phase constant and minimize over the amplitude and constant offset. In step 3) we hold the amplitude and offset constant and optimize over just the phase shift. Because of the constraints imposed at each step, the iterates $X(k)$ are assured of being PSD matrices. Since the cost can only decrease from iteration to iteration and it is bounded below, it must converge. Unfortunately, since the cost is not convex in the β_i, it is difficult to prove that the algorithm will always converge to the minimizer of Problem 3. Experience has shown

superior convergence properties in practice however.

Comments

Here we offer some comments about the various algorithms. We discuss their requirements and properties, possible extensions, and connections to other work.

The successive hyperplane algorithm 1 requires the repeated solution of a least squares problem with linear inequality constraints, termed problem LSI [55]. At each stage, however the major part of the constraint set has remained unchanged, as we have simply added additional constraints to the set. It seems that there should be some way to use our solution to the previous problem to help our solution to the augmented problem. This observation raises the question as to whether some recursive form of solution to problem LSI could be developed, at least in terms of the constraint set. As this step of our algorithm becomes the most computationally burdensome (due to its increasing dimension), such a recursive formulation would greatly speed up computation. We could simply update our solution as more constraints are generated.

Another algorithm that we investigated avoided using explicit gradient information (like the successive halfspace algorithm) and involved a successive rotation of the coordinate frame to attempt to iteratively diagonalize the solution. In the transformed diagonal space, the PSD constraint becomes a trivial one. Unfortunately, we did not have much success with the approach. It is interesting for its use of the underlying geometry, however, and so we have included it as Appendix 3-H.

The gradient based methods of Algorithms 2 and 3, being steepest descent algorithms, suffer from the poor terminal convergence rates that are typical of such approaches [63, 65, 66, 64]. This does not seem to be too much of a problem for the $n = 2$ case but does appear bothersome when $n = 3$. We have made little effort to optimize the algorithms from this standpoint. Rather our goal has been to illustrate the possibility of exploiting the special structure of the problem in the estimation of a PSD symmetric matrix. More efficient variants of these methods, even using the given expressions for the gradients, undoubtedly exist. For example, we have made no attempt to make use of curvature information, as would be provided by $\nabla^2 C$.

Little other work presently exists on solving problems with a semi-definite constraint on the solution, such as we have considered in Problem 3.12. The paper by Fletcher [48] is one such work in this vein. The primary interest in [48], however, is

on solving problems of the form

$$\begin{array}{ll} \text{maximize} & \|v\|_1, \qquad v \in R^n \\ \text{subject to} & S - \mathbf{diag}(v_i) \geq 0, \\ & v_i \geq 0 \end{array}$$

where **diag** denotes the diagonal matrix of the argument. The algorithm of the paper is a gradient based one, but not very intuitive from a geometric standpoint; and it has proved difficult to understand and apply.

Another paper with similar interests is by Allwright [67], and is more closely aligned to the present work. The interest in [67] is on solving a problem of the form:

$$\min_{X=X^T \geq 0} \|F - XG\|_F \tag{3.22}$$

where F and G are general matrices. Again, the approach taken has also proved difficult to understand because not much interpretation is provided for the various steps prescribed in the algorithm. It appears that the author of [67] does not impose the PSD constraint in the original space, but instead transforms the problem to a space where the minimization is straightforward but the PSD constraint is mapped to an equivalent warped constraint set, much as was discussed in connection with Figure 3-2. He uses the $\mathbf{vec}(X)$ formalism and a sort of elimination matrix to transform (3.22) to an equivalent vector problem given by

$$\min_{k \in F(\text{PSD})} \|u - k\|$$

where u and k are transformed variables and $F(\text{PSD})$ is some transformation of the PSD set. It is difficult to discern exactly how the PSD constraints are applied in the problem, as there are a number of layers of transformations and auxiliary variables.

3.4 Extensions

3.4.1 Interval Matrix Constraints

In this section we briefly consider how to generalize the results we have presented in this chapter to include a broader class of matrices than those with the PSD property. We demonstrate that most of the insights, representations, and algorithms presented

are easily extendible to sets of symmetric matrices lying in a *prescribed interval.* Consider the set of symmetric matrices defined by:

$$\overline{\underline{X}} = \{X | \overline{X} \geq X \geq \underline{X}\} \tag{3.23}$$

where by the inequalities $\overline{X} \geq X$ and $X \geq \underline{X}$ we mean that the matrices $(\overline{X} - X)$ and $(X - \underline{X})$ are PSD matrices. We term such a set, whose elements are constrained to lie in a specified matrix interval, a set of *interval matrices.* For the elements X of the PSD set we had the condition that $u^T X u \geq 0$, $\forall\ u^T u = 1$. Here, a matrix X is a member of a set of interval matrices $\overline{\underline{X}}$ in the sense we have defined if and only if

$$u^T \overline{X} u \geq u^T X u \geq u^T \underline{X} u, \quad \forall\ u^T u = 1$$

We may now immediately use our development of Section 2.3, particularly Section 2.3.1, to obtain the following characterization of a symmetric interval matrix set:

$$\overline{\underline{X}} = \bigcap_{\substack{N \in \mathrm{PSD}_n^{(1)} \\ \|N\|_F = 1}} \mathcal{HS}(N, \langle N, \underline{X} \rangle) \cap \mathcal{HS}(-N, -\langle N, \overline{X} \rangle) \tag{3.24}$$

We can see that this set is just the intersection of *two* cones, each a copy of the PSD cone. The first cone $\cap\, \mathcal{HS}(N, \langle N, \underline{X} \rangle)$ is just the PSD cone shifted by the vector $\underline{X}$. The second cone $\cap\, \mathcal{HS}(-N, -\langle N, \overline{X} \rangle)$ is a flipped version of the PSD cone translated to the point $\overline{X}$. Thus the overall set is the intersection of two offset cones.

Eigenvalue Interval Set

For example, suppose we choose the bounding matrices $\overline{X}$ and $\underline{X}$ so that $\overline{X} = \overline{\lambda} I$ and $\underline{X} = \underline{\lambda} I$, with $\overline{\lambda}$ and $\underline{\lambda}$ as nonnegative scalars. The elements of the corresponding set of matrices $\overline{\underline{X}}$ are constrained to have eigenvalues $\lambda_i(X)$ in the interval $\overline{\lambda} \geq \lambda_i(X) \geq \underline{\lambda}$. We term such a set an *eigenvalue interval* set, notation $\overline{\underline{\mathrm{EIG}}}$. For this case it follows that $\langle N, \overline{X} \rangle = \overline{\lambda}$ and $\langle N, \underline{X} \rangle = \underline{\lambda}$ and the translations are along the direction I. The overall set $\overline{\underline{\mathrm{EIG}}}$ is illustrated for the $n = 2$ case in Figure 3-3, where we have cut away part of the boundary to reveal the interior.

Note that for this $n = 2$ case the intersection of the two cones represents the set of matrices with one eigenvalue at the maximum value and one at the minimum value. Since there are only two eigenvalues for this case, the resulting circle represents the set of 2×2 matrices with *prescribed* eigenvalues. In particular, notice that these circles

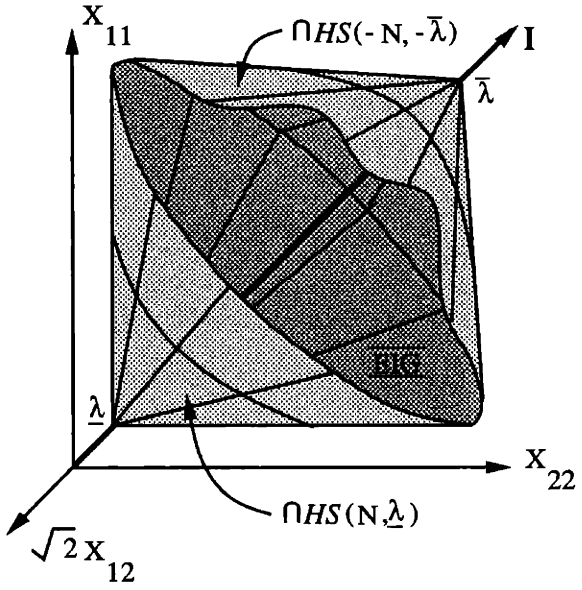

Figure 3-3: Illustration of eigenvalue interval set, $n = 2$.

must be concentric as seen along the identity I direction. The matrices with greater extremes in their eigenvalues lie on circles farther out from the identity direction than those whose eigenvalues are closer together. In fact, the set of matrices with a given difference between their eigenvalues lie on the surface of a cylinder of a fixed radius with the identity matrix I along the central axis.

Constrained Resonstruction

An interesting problem is the reconstruction of symmetric matrices under interval constraints of the form given in (3.23). In particular consider the following problem:

Problem 6 (Interval Matrix Reconstruction) *Determine the $n \times n$ symmetric matrix $X_{\underline{\overline{X}}}$ that solves the following constrained linear least square error (LLSE) problem:*

$$\min_{X \in \underline{\overline{X}}} \sum_{i=1}^{q} \left\| Y_i - A_i^T X A_i \right\|_F^2$$

where each Y_i is an $m_i \times m_i$ symmetric matrix, each A_i is an $n_i \times m_i$ real matrix, and $\underline{\overline{X}}$ is a set of interval matrices as defined in (3.23) or (3.24).

The characterization of (3.24) enables us to apply the set approximation insights developed in Section 3.3.2 to this interval matrix case. For example, we may develop external approximations as in Lemma 2 by replacing the halfspaces $\mathcal{HS}(N_i, 0)$ used

there with the sets $\mathcal{HS}(N_i, \langle N_i, \underline{X}\rangle) \cap \mathcal{HS}(-N_i, -\langle N_i, \overline{X}\rangle)$. Once the limits $\underline{X}$ and $\overline{X}$ are fixed these sets are completely determined by their associated set of normals $\{N\}$. Note that the resulting approximations are still defined by a group of *linear inequality constraints*, as before. Let $\underline{\widetilde{X}}$ denote such an approximating set. We have a particular interest in generalizing the iterative halfspace algorithm of Section 3.3.3. Such an extension is provided in the following:

Algorithm 5 (Interval Matrix Successive Halfspace)

Step 1) *Choose an initial external approximating set of unit rank 1 hyperplane normals* $\{N_i\}_0$, *thus defining* $\underline{\widetilde{X}}_0$. *Set* $k = 0$.

Step 2) *Solve the following to obtain* $X(k)$:

$$X(k) = \arg\min_{X \in \underline{\widetilde{X}}_k} \sum_{i=1}^{q} \left\| Y_i - A_i^T X A_i \right\|_F^2$$

where

$$\underline{\widetilde{X}}_k = \bigcap_{N \in \{N_i\}_k} \mathcal{HS}(N, \langle N, \underline{X}\rangle) \cap \mathcal{HS}(-N, -\langle N, \overline{X}\rangle)$$

Step 3) *For the lower bound, find all eigenvalues* $\underline{\lambda}_j$ *and corresponding eigenvectors* $\underline{v}_j$ *of* $[X(k) - \underline{X}]$ *such that* $\underline{\lambda}_j \leq 0$.

Step 4) *For the upper bound, find all eigenvalues* $\overline{\lambda}_j$ *and corresponding eigenvectors* $\overline{v}_j$ *of* $[\overline{X} - X(k)]$ *such that* $\overline{\lambda}_j \leq 0$. *If there are no* $\underline{\lambda}_j$ *or* $\overline{\lambda}_j$ *the solution is optimal: STOP.*

Step 5) *Update the approximating set* $\underline{\widetilde{X}}_k$ *by augmenting the associated set* $\{N_i\}_k$ *with rank 1 normals corresponding to the forbidden eigenvectors found in Steps 3) and 4):* $\{N_i\}_{k+1} = \{N_i\}_k \cup \{\underline{v}_j \underline{v}_j^T\} \cup \{\overline{v}_j \overline{v}_j^T\}$.

Step 6) *Goto Step 2).*

Similar types of convergence arguments can be made for this algorithm as for the original successive halfspace algorithm. As for that algorithm, the present one also keeps adding halfspaces needed to build up a successively better local approximation to the set $\underline{X}$.

Eigenvalue Interval Algorithm

In the case of an eigenvalue interval constraint set $\underline{\overline{\text{EIG}}}$, we may also adapt the gradient projection Algorithm 3.17 to solve Problem 6. Instead of projecting back onto the PSD set, as we did before, we project onto the eigenvalue interval set $\underline{\overline{\text{EIG}}}$ at each iteration. Given an arbitrary symmetric matrix Z, the nearest matrix $[Z]^{\underline{\overline{\text{EIG}}}}$ in the Frobenius norm to the set $\underline{\overline{\text{EIG}}}$ is given by:

$$[Z]^{\underline{\overline{\text{EIG}}}} = \sum_{\overline{\lambda} \geq \lambda_i(Z) \geq \underline{\lambda}} \lambda_i(Z)\ v_i v_i^T + \sum_{\lambda_i(Z) \leq \underline{\lambda}} \underline{\lambda} v_i v_i^T + \sum_{\lambda_i(Z) \geq \overline{\lambda}} \overline{\lambda} v_i v_i^T \tag{3.25}$$

where $\lambda_i(Z)$ are the eigenvalues of Z and v_i are the corresponding eigenvectors. Thus $[Z]^{\underline{\overline{\text{EIG}}}}$ is obtained from Z simply setting any eigenvalues of Z which are outside of the required range to the nearest limit. Now notice that the cost function $C(X) = \sum_{i=1}^{q} \left\| Y_i - A_i^T X A_i \right\|_F^2$ has remained the same, so the expressions for $\nabla C(X)$ derived earlier are applicable here. With these observations we have the following generalization of the PSD gradient projection algorithm:

Algorithm 6 (Eigenvalue Interval Gradient Projection)

Step 1) *Choose an initial guess* $X(0)$. *Set* $k = 0$.

Step 2) *Evaluate* $\nabla C(X(k))$.

Step 3) *Take a step.* $W(k) = X(k) - \gamma \nabla C(X(k))$.

Step 4) *Project back onto the PSD set to obtain the next iterate.* $X(k+1) = [W(k)]^{\underline{\overline{\text{EIG}}}}$.

Step 5) *Goto 2).*

The corresponding projection algorithm for the general matrix interval constraint set $\underline{\overline{X}}$ requires a simple expression for the projection onto the set $\underline{\overline{X}}$, similar to (3.25).

Summary

In summary, these extensions to our work allow us to apply constrained symmetric matrix reconstruction to a much larger class of problems. We may now easily consider problems of the form (2) involving a symmetric matrix with interval matrix constraints. Such constraints might reflect prior physical information regarding the object under consideration. For example, in the ellipsoid reconstruction problem

we consider, such interval constraints directly reflect inner and outer bounds on the shape of the ellipsoid. We may thus incorporate prior knowledge regarding the inner and outer limits to the desired ellipsoid directly into the reconstruction process. We postpone further discussion of these ideas until Chapter 5.

3.4.2 Other Linear Problems

As discussed at the end of Section 2.2, since we may represent any linear operator $Y = \mathcal{L}(X)$ in the form $\mathbf{y} = \mathbf{A}x$ and since the PSD approximations and constraints we have developed are linear functions of the representation x, most of our algorithms and approximations are immediately applicable to the larger class of these linear problems involving a symmetric matrix X. Only the direct form of the cost minimization and gradient expressions for the gradient algorithms 2, 3, and 6 need to be changed. The expression of these algorithms in the symmetric vector space S is already in terms of the matrix representation $\mathbf{A}$ of the linear operator $\mathcal{L}(\cdot)$ and thus once the appropriate matrix representation of the operator $\mathcal{L}(X)$ is found, these expressions *are* directly applicable. In short, any linear problem with a symmetric matrix argument may be treated using the results of this chapter, e.g. problems of the form $Y_i = A_i^T X B_i$, $Y_i = A_i^T X + X B_i$, etc.

With slightly more work, nonlinear problems in X can also benefit from the insights of this chapter. For example, consider the halfspace Algorithm 1. We could replace Step 2) with the minimization of *any* cost function $\tilde{C}(X)$ (possibly nonlinear) subject to the applied linear inequality constraints. If solving this new linearly constrained problem is practicable as a replacement for Step 2), then the modified algorithm gives a simple way of minimizing $\tilde{C}(X)$ over the set of PSD matrices. The only requirement is to be able to solve the minimization subject to a series of linear constraints. Very often this can be done. Naturally, matrix interval variants are also possible, resulting in constrained reconstructions in which the solution matrix is contained in a given interval.

3.5 Numerical Experiments

In this section we provide some numerical experiments illustrating the approximations and reconstruction techniques developed above. We start by introducing the example data sets. Next a brief demonstration of the non-iterative external and internal

X	A_1	A_2	A_3	A_4
$\begin{bmatrix} 2.125 & 1.875 \\ 1.875 & 2.125 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix}$	$\begin{bmatrix} 0.707 \\ -0.707 \end{bmatrix}$
Eigenvalues of X	Y_1	Y_2	Y_3	Y_4
4, 1/4	3.049	0.311	4.035	−1.558

Figure 3-4: Data of Example 1.

approximate reconstructions of Problem 4 is given. These approximations use the sets developed in Sections 2.4.2 and 2.4.3. We show how increasing the number of faces in the approximating set improves the estimate. Then we demonstrate the iterative algorithms developed in Section 3.3.3. Finally we conclude by demonstrating iterative extensions to constrained eigenvalue interval matrix reconstruction.

3.5.1 Examples

We consider two examples, both of them with $n = 2$ and $m = 1$ corresponding to a 2×2 matrix with scalar observations. We use this case because of its ease of visualization. The results we have shown (with the exception of the projection space reconstruction method) hold for arbitrary combinations of dimensions. An easy and a hard example are used for comparison. Both problems use the same underlying matrix, but differ in the geometry of their observations. The first example uses well spaced observations of both eigen-axes while the second has nearly dependent observations of the smaller eigen-axis. In both cases we have added zero mean Gaussian noise of variance 1 (notation $N(0,1)$) to the observations, so that our observations were obtained as $Y_i = A_i^T X A_i + W_i$, where $W_i \sim N(0,1)$.

Example 1

The first example that we use is an example of an "easy" problem. By this we mean that the geometry of the observations is well spaced and there are redundant observations. In Figure 3-4 we give the numerical particulars of this example. Notice in particular that, due to noise, observation Y_4 is negative even though the eigenvalues of X are positive. This observation was along the smaller eigen-direction of the matrix. In Figure 3-5 we show the projection subspaces together with the eigen-directions and

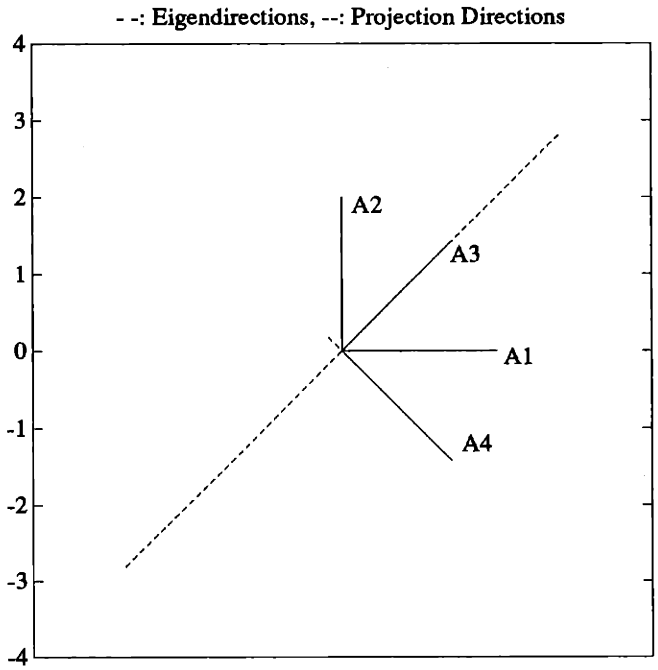

Figure 3-5: Geometry of Example 1.

with their lengths scaled according to the corresponding eigenvalues. As can be seen, the matrix is quite eccentric: one eigenvalue is much larger than the other.

Example 2

The second example we use is an example of a harder problem. The underlying matrix is the same as that in Example 1, but only the minimal number of observations are used and the geometry of the observations is such that two projections are nearly redundant. In Figure 3-6 we give the numerical particulars of Example 2. Notice that views 2 and 3 are nearly the same and are aligned with the small eigenvalue. Again, due to noise, observation Y_2 is negative even though the eigenvalues of X are positive. This observation was along the smaller eigen-direction of the matrix. In Figure 3-7 we show the projection subspaces together with the eigen-directions with their lengths scaled to the corresponding eigenvalues.

X	A_1	A_2	A_3
$\begin{bmatrix} 2.125 & 1.875 \\ 1.875 & 2.125 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0.707 \\ -0.707 \end{bmatrix}$	$\begin{bmatrix} 0.8 \\ -0.6 \end{bmatrix}$
Eigenvalues of X	Y_1	Y_2	Y_3
4, 1/4	2.854	−2.127	0.051

Figure 3-6: Data of Example 2.

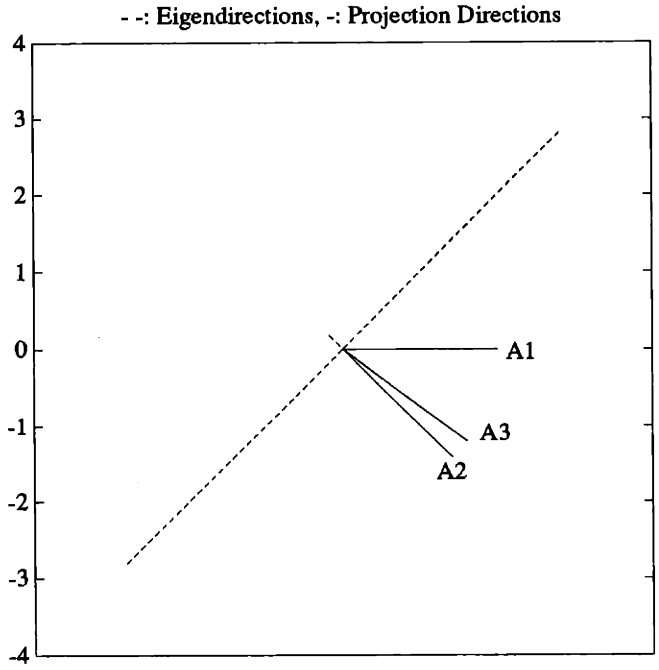

Figure 3-7: Geometry of Example 2.

3.5.2 Non-Iterative Approximations and Reconstructions

Here we demonstrate some of the non-iterative interior and exterior approximations and reconstructions developed in Sections 2.4.2 and 2.4.3. In Figure 3-8 we show the estimates and cost generated for Example 1 by the unconstrained LLSE solution X_{LLSE}, the minimum cost PSD solution X_{PSD}, the nearest PSD matrix to the LLSE solution X_{near}, and the Geršgorin polyhedral cone solution. In Figure 3-9 are the same estimates for Example 2.

Note in both cases that the unconstrained LLSE solution is not positive semi-

Approximation	Estimate $\hat{X}$	Eigenvalues	Cost $C(\hat{X})$
Unconstrained LLSE	$\begin{bmatrix} 2.828 & 2.796 \\ 2.796 & 0.090 \end{bmatrix}$	4.573, −1.654	0.195
Exact PSD Constrained	$\begin{bmatrix} 2.895 & 1.806 \\ 1.806 & 1.127 \end{bmatrix}$	4.021, 0	3.844
Nearest PSD to LLSE	$\begin{bmatrix} 3.292 & 2.053 \\ 2.053 & 1.281 \end{bmatrix}$	4.573, 0	4.300
Geršgorin Cone PSD	$\begin{bmatrix} 2.287 & 1.714 \\ 1.714 & 1.714 \end{bmatrix}$	3.738, 0.263	6.054

Figure 3-8: Comparison of different estimates for Example 1.

Approximation	Estimate $\hat{X}$	Eigenvalues	Cost $C(\hat{X})$
Unconstrained LLSE	$\begin{bmatrix} 2.854 & -3.266 \\ -3.266 & -13.64 \end{bmatrix}$	3.477, −14.265	0
Exact PSD Constrained	$\begin{bmatrix} 2.851 & 2.870 \\ 2.870 & 2.888 \end{bmatrix}$	5.740, 0	4.530
Nearest PSD to LLSE	$\begin{bmatrix} 3.355 & -0.640 \\ -0.640 & 0.122 \end{bmatrix}$	3.477, 0	28.143
Geršgorin Cone PSD	$\begin{bmatrix} 2.851 & 2.851 \\ 2.851 & 2.851 \end{bmatrix}$	5.702, 0	4.530

Figure 3-9: Comparison of different estimates for Example 2.

definite, even though the underlying matrix is, due to the noise in the observations. Also, note that the cost $C(X)$ is not comparable between Examples 1 and 2. Recall that the cost is just the sum of the residuals squared. The cost of the unconstrained solution in Example 2 is 0 because we have just the minimum number of views to uniquely determine a solution, so the residual will be 0 for this case.

Uniform Polyhedral Cones

Here we examine approximations based on uniform polyhedral cones, as shown in Figures 2-6 and 2-12. We demonstrate both the exterior approximations based on Result 11 and the corresponding interior sets, based on Result 14. Recall that these latter sets are obtained from the former exterior ones by moving the support halfspaces inward toward the central direction of the cone. In all cases we move them by the minimum amount necessary to obtain an interior approximation (corresponding to choosing $t = t^\star$ in Result 14). Since we use a uniform spacing of the halfspaces, or equivalently from Result 11, a uniform tessellation of the unit circle, the only free parameter is the number of faces of the approximation. First we show the exterior approximations and then the corresponding interior approximations for each example.

In Figure 3-10 we have the exterior approximations for Example 1, while Figure 3-11 shows the corresponding interior approximations for this case. Note that the interior approximation with 4 faces corresponds to a rotated Geršgorin polyhedron; both have 4 faces. Figure 3-12 shows the external polyhedral cone results for Example 2 while Figure 3-13 shows the corresponding internal approximations for this example.

The exterior approximations approach the optimal PSD constrained cost from below (since the exterior constraint sets are *larger* than the PSD set) while the corresponding interior approximations approach this cost from above. In Figure 3-14 we plot both these costs verses the number of facets in the approximation for Example 2. Note the oscillations in the cost. This effect is due to the fact that one edge of the polyhedral cone is always fixed, so that as more faces are added, the points of contact of the approximation with the boundary of the PSD set will effectively sweep along the PSD cone boundary. Since the optimal PSD constrained solution is *on* the boundary, at those (internal or external) face combinations where the contact points happen to match the PSD solution point the cost will drop or rise to its minimum constrained value. The curves do not cross this line. In Figure 3-15 we have plotted the difference in these interior and exterior costs, which shows clearly the fact that the difference between the interior and exterior approximations tends to zero from above as the number of faces increases.

External Polyghedral Cones

Number of Faces	Estimate $\hat{X}$	Eigenvalues	Cost $C(\hat{X})$
3	$\begin{bmatrix} 2.828 & 1.845 \\ 1.845 & 1.188 \end{bmatrix}$	4.028, −0.011	3.811
4	$\begin{bmatrix} 3.274 & 1.905 \\ 1.905 & 0.536 \end{bmatrix}$	4.251, −0.441	2.579
6	$\begin{bmatrix} 2.828 & 1.845 \\ 1.845 & 1.188 \end{bmatrix}$	4.679, 0	3.812
8	$\begin{bmatrix} 2.801 & 1.981 \\ 1.981 & 1.160 \end{bmatrix}$	4.125, −0.163	3.215
10	$\begin{bmatrix} 3.035 & 1.767 \\ 1.767 & 0.966 \end{bmatrix}$	4.048, −0.047	3.709

Figure 3-10: Comparison of external polyhedral cone estimates for Example 1.

Internal Polyhedral Cones

Number of Faces	Estimate $\hat{X}$	Eigenvalues	Cost $C(\hat{X})$
3	$\begin{bmatrix} 2.300 & 0.967 \\ 0.967 & 1.675 \end{bmatrix}$	3.004, 0.971	10.237
4	$\begin{bmatrix} 3.327 & 1.385 \\ 1.385 & 0.589 \end{bmatrix}$	3.906, 0.011	5.177
6	$\begin{bmatrix} 2.739 & 1.636 \\ 1.636 & 1.340 \end{bmatrix}$	4.679, 0	5.133
8	$\begin{bmatrix} 2.777 & 1.856 \\ 1.856 & 1.240 \end{bmatrix}$	4.017, 0	3.893
10	$\begin{bmatrix} 3.023 & 1.686 \\ 1.686 & 1.006 \end{bmatrix}$	3.979, 0.049	4.150

Figure 3-11: Corresponding internal polyhedral cone estimates for Example 1.

External Polyhedral Cones

Number of Faces	Estimate $\hat{X}$	Eigenvalues	Cost $C(\hat{X})$
3	$\begin{bmatrix} 2.940 & 4.459 \\ 4.459 & 4.169 \end{bmatrix}$	8.055, −0.947	2.404
4	$\begin{bmatrix} 2.854 & 3.116 \\ 3.116 & 3.378 \end{bmatrix}$	6.243, −0.011	4.526
6	$\begin{bmatrix} 3.068 & 3.542 \\ 3.542 & 3.068 \end{bmatrix}$	6.610, −0.475	2.925
8	$\begin{bmatrix} 2.854 & 3.116 \\ 3.116 & 3.378 \end{bmatrix}$	6.243, −0.011	4.526
10	$\begin{bmatrix} 2.955 & 3.107 \\ 3.107 & 2.955 \end{bmatrix}$	6.061, −0.152	3.919

Figure 3-12: Comparison of external polyhedral cone estimates for Example 2.

Internal Polyhedral Cones

Number of Faces	Estimate $\hat{X}$	Eigenvalues	Cost $C(\hat{X})$
3	$\begin{bmatrix} 2.127 & 3.684 \\ 3.684 & 6.381 \end{bmatrix}$	8.507, 0	7.809
4	$\begin{bmatrix} 2.232 & 0.924 \\ 0.924 & 0.383 \end{bmatrix}$	2.615, 0	7.082
6	$\begin{bmatrix} 2.851 & 2.851 \\ 2.851 & 2.851 \end{bmatrix}$	5.702, 0	4.530
8	$\begin{bmatrix} 2.669 & 1.783 \\ 1.783 & 1.192 \end{bmatrix}$	3.860, 0	5.347
10	$\begin{bmatrix} 2.851 & 2.851 \\ 2.851 & 2.851 \end{bmatrix}$	5.702, 0	4.530

Figure 3-13: Corresponding internal polyhedral cone estimates for Example 2.

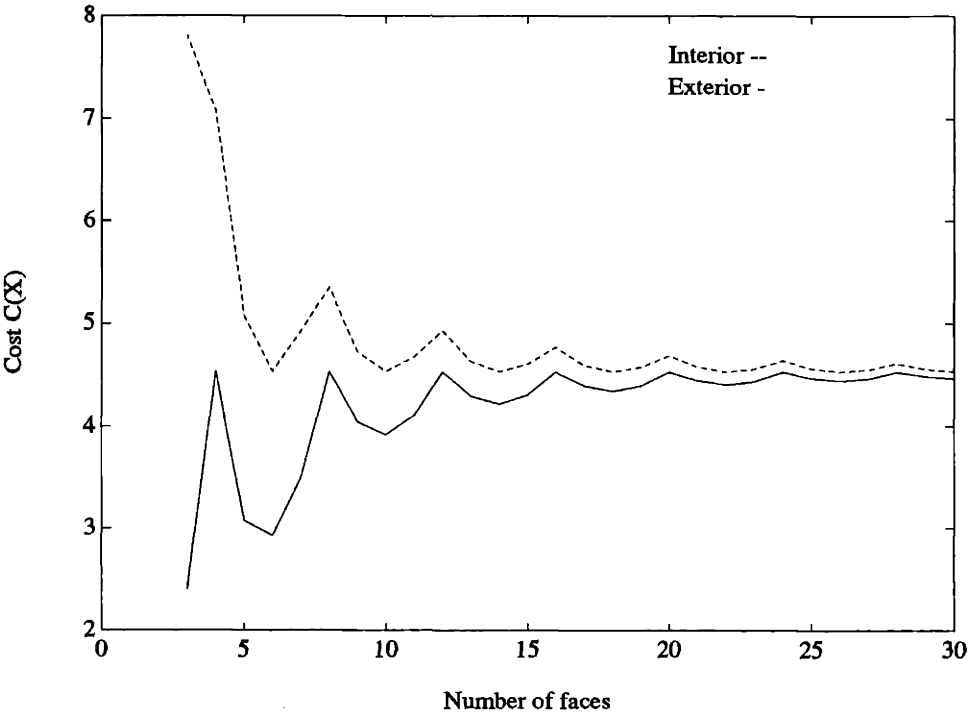

Figure 3-14: Comparison of exterior and interior costs for Example 2.

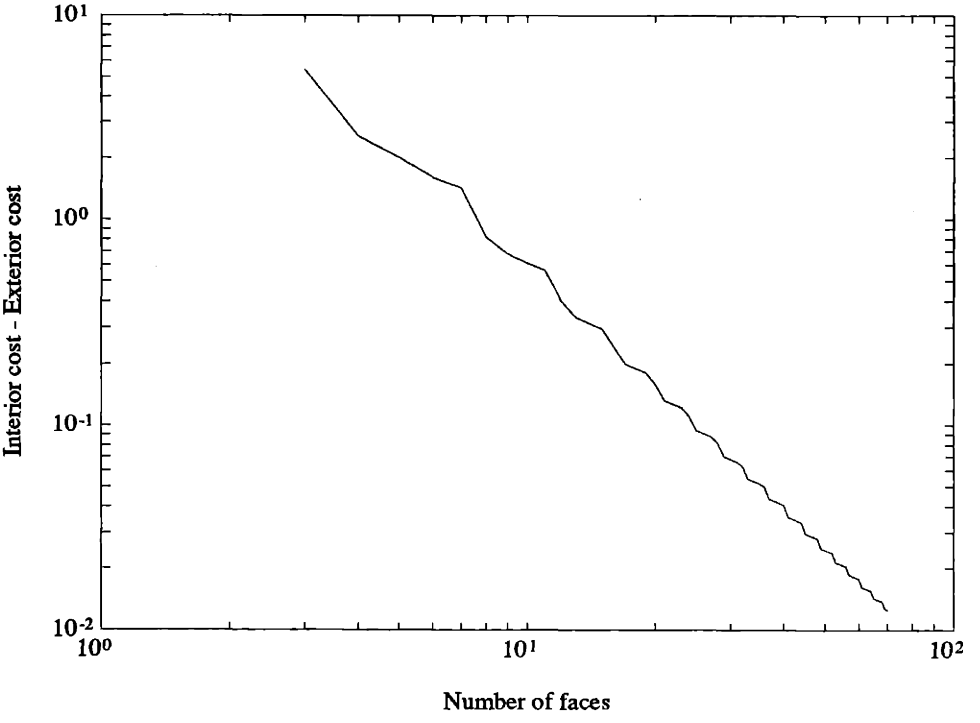

Figure 3-15: Difference between interior and exterior costs for Example 2.

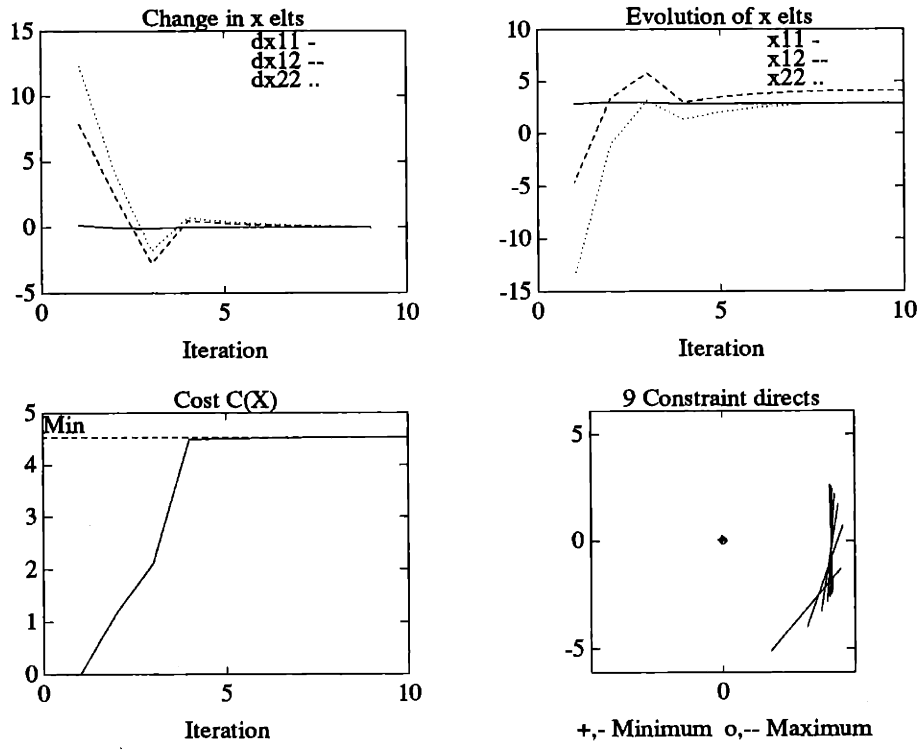

Figure 3-16: Successive halfspace algorithm applied to Example 2.

3.5.3 Iterative reconstructions

Now we consider the iterative methods of reconstruction presented in Section 3.3.3. All these methods attempt to find a solution to the PSD constrained Problem 3. We only show results for Example 2, as they are illustrative of the performance of the algorithms.

Successive Halfspace

In Figure 3-16 is shown the operation of the successive halfspace method of Algorithm 1. We have displayed both the change (first order difference) in the elements in the symmetric space and their evolution along the top row. On the bottom left the cost of the estimate at each iteration is displayed along with the minimum cost of the optimal PSD constrained solution. On the bottom right is shown the resulting PSD set approximation from a similar vantage point to that given on the right of Figure 2-6a[1]. The displayed lines are the intersection of the successive bounding

[1]The first coordinate axis in the plot of Figure 3-16 is along the positive X_{12} axis, so that this view is from the origin looking out along the positive I direction

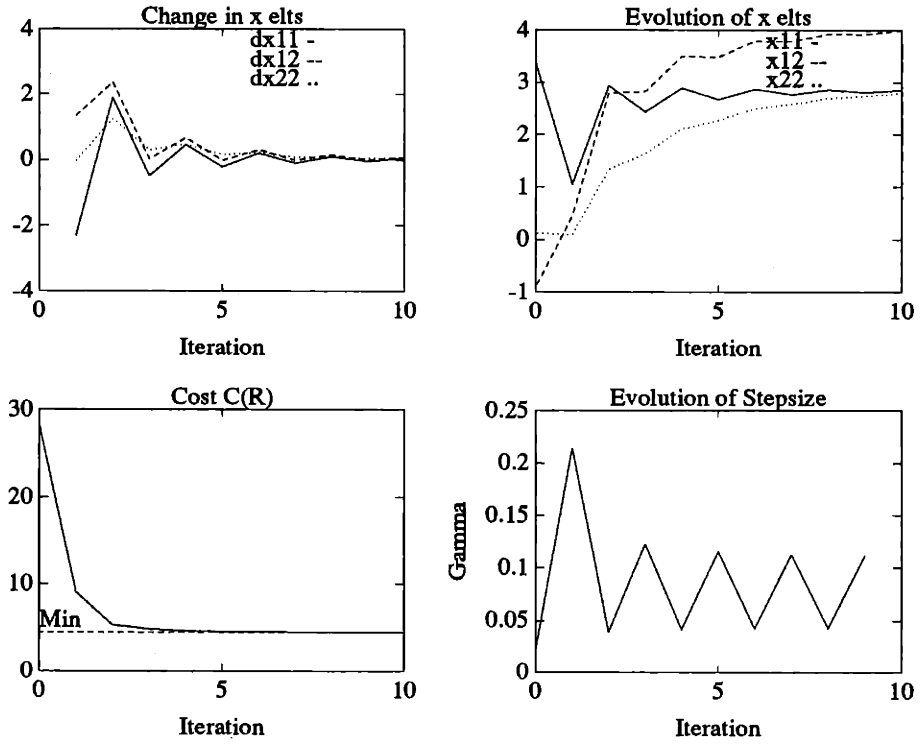

Figure 3-17: Unconstrained gradient descent algorithm applied to Example 2.

hyperplanes with the hyperplane $\mathcal{H}(I, \alpha)$, for a fixed scalar α. As can be seen, the algorithm is building an approximation of the PSD cone at only a single place. Since the algorithm approximates the PSD cone from the outside, the cost rises to meet the minimum PSD constrained cost from below. The final estimate after 10 iterations is given by:

$$\hat{X} = \begin{bmatrix} 2.851 & 2.876 \\ 2.876 & 2.900 \end{bmatrix}$$

The eigenvalues of $\hat{X}$ are given by 0 and 5.75, so that the algorithm has indeed produced a PSD estimate.

Unconstrained Steepest Descent on a Square Root

In Figure 3-17 we show the output of the unconstrained steepest descent method of Algorithm 2, which uses a square root of the matrix. Here we have used the variable stepsize rule, whereby we choose the stepsize at each iteration to minimize the cost at that iteration. Again, both the change (first order difference) in the elements in the symmetric space and their evolution are displayed along the top row. Notice

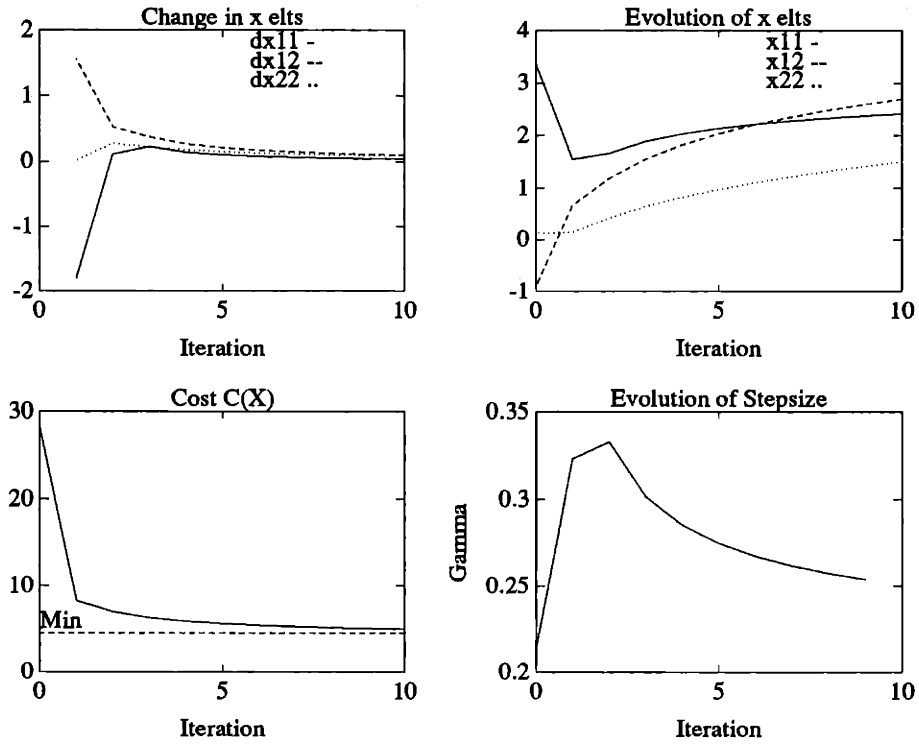

Figure 3-18: Gradient projection algorithm applied to Example 2.

that after 10 iterations the elements are still moving towards their final values. On the bottom left the cost of the estimate at each iteration is displayed along with the minimum cost of the optimal PSD constrained solution. On the bottom right is the stepsize chosen at each iteration. After 10 iterations the estimate is given by:

$$\hat{X} = \begin{bmatrix} 2.856 & 2.823 \\ 2.823 & 2.792 \end{bmatrix}$$

The eigenvalues of $\hat{X}$ are given by 0 and 5.647 so the matrix is again PSD.

Gradient Projection on X

In Figure 3-18 we show the gradient projection method of Algorithm 3. We have again used the variable stepsize rule, choosing the stepsize at each iteration to minimize the cost at that iteration. Again, both the change (first order difference) in the elements in the symmetric space and their evolution is displayed along the top row. Notice that after 10 iterations the elements are still moving towards their final values. On the bottom left the cost of the estimate at each iteration is displayed along with the

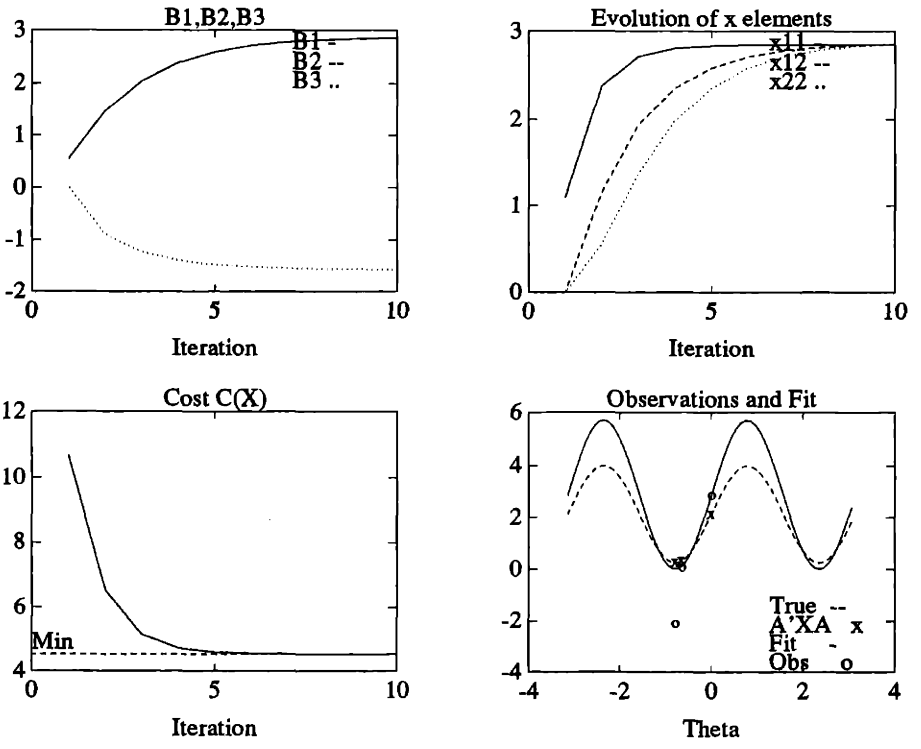

Figure 3-19: Sine fitting algorithm applied to Example 2.

minimum cost of the optimal PSD constrained solution. On the bottom right is the stepsize chosen at each iteration. After 10 iterations the estimate is given by:

$$\hat{X} = \begin{bmatrix} 2.418 & 1.908 \\ 1.908 & 1.505 \end{bmatrix}$$

The eigenvalues of this estimate are given by 0 and 3.923 so the estimate is again a PSD matrix. By construction, the two gradient based algorithms produce estimates which are PSD matrices at every iteration, unlike the successive halfspace method of Algorithm 1.

Projection Parameter Space

In Figure 3-19 we show the positive sine fitting parameter space approach of Algorithm 4. In the top left corner the evolution of the projection space parameters of the sinusoid β_1, β_2, and β_3 are shown. Recall that in the noise free case

$$Y_i(\theta_i) = \beta_1 + \beta_2 \cos(2\theta_i + \beta_3)$$

Notice that β_1 and β_2 are the same, since the cosine touches the θ axis. In the top right corner is shown the corresponding evolution of the original symmetric space matrix elements. The bottom left corner shows the cost of the estimate at each iteration, along with the minimum cost of the optimal PSD constrained solution. On the bottom right we have shown the resulting positive cosine fit as a solid line, the true underlying cosine as a broken line, the corresponding noisy observations as circles (o), and the observations that would have been obtained with no noise ($= A^T X A$) as crosses (x). Even though the noise free observations are positive, the addition of noise has resulted in a large negative value. After 10 iterations the estimate is given by:

$$\hat{X} = \begin{bmatrix} 2.851 & 2.855 \\ 2.855 & 2.859 \end{bmatrix}$$

The eigenvalues of this estimate are given by 0 and 5.71 so that the estimate $\hat{X}$ is indeed a PSD matrix.

Interval Matrix Constrained Reconstruction

For interest, we now demonstrate the effect of including matrix interval constraints in the reconstruction process. Suppose that we had prior information that the underlying matrix X lay in the range $\overline{X} \geq X \geq \underline{X}$, where the bounds $\overline{X}$ and $\underline{X}$ are given by:

$$\overline{X} = \begin{bmatrix} 3 & 3 \\ 3 & 6 \end{bmatrix} \quad \underline{X} = \begin{bmatrix} .46 & .16 \\ .16 & .13 \end{bmatrix}$$

It can be verified that the true X is indeed in this range. Using this constraint in the interval matrix successive halfspace method of Algorithm 5 yields the results shown in Figure 3-20. Again, both the change (first order difference) in the elements in the symmetric space and their evolution are displayed along the top row. On the bottom left the cost at each iteration is shown. On the bottom right is again shown the resulting PSD set approximation from a similar vantage point to that given on the right of Figure 2-6a. There are now two cones being approximated, corresponding to the upper and lower bound. The circle in the bottom right of Figure 3-20 corresponds to the tip of the upper bound cone. The dotted lines are the corresponding bounding hyperplanes for this cone chosen by the algorithm. Similarly, the cross corresponds to the tip of the lower bound cone and the solid lines the associated hyperplanes for this cone. Since the upper and lower bounds $\overline{X}$ and $\underline{X}$ are not along the I direction (which

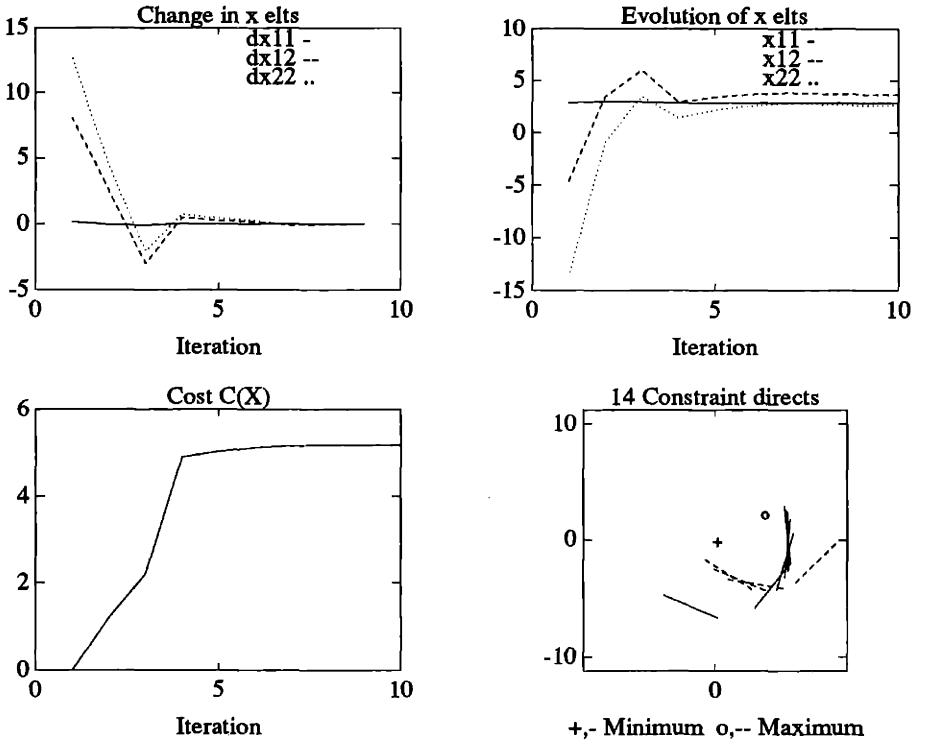

Figure 3-20: Successive halfspace interval algorithm applied to Example 2.

is going into the figure), the tips are shifted from the origin. After 10 iterations we obtain for the estimate:

$$\hat{X} = \begin{bmatrix} 2.845 & 2.585 \\ 2.585 & 2.595 \end{bmatrix}.$$

The eigenvalues of this matrix are .132 and 5.31, so the estimate is certainly a PSD matrix.

Note that the final cost for this estimate is higher than for the reconstructions that are only PSD constrained, as expected. However, if we examine the percentage reconstruction error, defined by:

$$\text{Percentage Error} \;= 100\frac{\|X_{\text{true}} - \hat{X}\|_F^2}{\|X_{\text{true}}\|_F^2}$$

where X_{true} is the true underlying symmetric matrix, we find that PSD constrained reconstruction has about a 20% error while the bounded eigenvalue reconstruction has only about a 10% error. Thus, not surprisingly, we actually do a better job of reconstructing the underlying generating matrix with the incorporation of prior information through constraints.

3.6 Conclusions

In this chapter we have examined the inverse problem of reconstructing a symmetric matrix X given one or more projection mapping observations $Y = A^T X A$. A complete solution to the unconstrained Problem 2, both for the consistent and the inconsistent cases was provided. The approach used involved the isomorphism defined in Chapter 2. The nature of this isomorphism assured the symmetry of all solutions and the vector space structure enabled us to use the insights and results of linear algebra. In addition to the above formalism, an interpretation and solution of the problem for the $n = 2$ case directly in the space of projections was given.

Next the issue of PSD constrained reconstruction was introduced. We showed how to couple the approximations of the PSD set developed in Chapter 2 with the unconstrained minimization problem above to yield constrained PSD symmetric reconstructions. First we used our PSD set approximations in a static, non-iterative way to provide approximations to the true minimizing PSD solution to our problem. The PSD approximations, defined as they were by sets of linear inequality constraints, yielded simple linearly constrained LLSE problems (problem LSI).

Iterative methods to solve this PSD constrained problem were then examined. A series of new algorithms was presented for this purpose. The successive halfspace algorithm used our understanding of the geometry of the PSD cone in a novel way to yield an algorithm assured of converging. Gradient based algorithms were also developed. The framework we used for imposing positive semi-definite constraints on the reconstructed solution was also extended to add arbitrary eigenvalue bounds to the answer, relating our class of problems to an even broader class. We concluded by presenting some numerical experiments demonstrating the approximations and algorithms developed throughout this chapter.

3-A Proof of Result 17

Necessity: If a solution of (2.3) exists, then, using the definitions in Chapter 2, with straightforward substitutions we may show that $A^T X A = Y$ is equivalent to:

$$\begin{bmatrix} \Sigma X_{11} \Sigma & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \tag{3.26}$$

Clearly we must have $Y_{11} = 0$ and $Y_{12} = 0$ for a solution to exist.

Sufficiency Since equation (2.3) is equivalent to (3.26), then (2.3) is consistent if (3.26) is. Now by inspection, (3.26) is consistent under the hypotheses of the result.

Finally, by construction we can see that the given set of solutions satifies (2.3).

3-B Derivation of normal equations

Here we will derive the equivalent normal equations for Problem 2 in the original space of symmetric matrices **S**. Recall, we are trying to solve the following problem:

$$\begin{aligned} &\min_{X=X^T} \sum_{i=1}^{q} \left\| Y_i - A_i^T X A_i \right\|_F^2 \\ &= \min_{X=X^T} \sum_{i=1}^{q} \mathsf{Tr}\left((Y_i - A_i^T X A_i)^T (Y_i - A_i^T X A_i) \right) \\ &= \min_{X=X^T} \sum_{i=1}^{q} \mathsf{Tr}\left(Y_i^T Y_i - A_i^T X A_i Y_i - Y_i A_i^T X A_i + A_i^T X A_i A_i^T X A_i \right) \end{aligned}$$

Now the solution of this equation must be a stationary point, so all solutions $\hat{X}$ must satisfy $\partial C(X)/\partial X|_{\hat{X}} = 0$ where $C(X)$ is the expression above. This gives:

$$\partial C(X)/\partial X = \sum_{i=1}^{q} \left(-2 A_i Y_i A_i^T + 2 A_i A_i^T X A_i A_i^T \right)$$

Or equivalently:

$$\sum_{i=1}^{q} A_i A_i^T \hat{X} A_i A_i^T = \sum_{i=1}^{q} A_i Y_i A_i^T \equiv Y \tag{3.27}$$

where we signify that the right hand side is completely determined by giving it the symbol Y. Thus we seek symmetric solutions of (3.27) given the symmetric matrix Y and the matrices A_i. This is the direct formulation of the problem in the orignal space, and (3.27) serves as the normal equations.

Note that when the A_i have orthonormal columns (and thus are also of full rank), (3.27) becomes:

$$\sum_{i=1}^{q} P_i \hat{X} P_i = \sum_{i=1}^{q} A_i Y_i A_i^T$$

where P_i is a projector [37] onto the subspace spanned by the columns of A_i. Further, if the A_i themselves are such projectors, so that $A_i A_i^T = A_i$, then this becomes:

$$\sum_{i=1}^{q} A_i \hat{X} A_i = \sum_{i=1}^{q} A_i Y_i A_i^T$$

and in addition, if the observations Y_i are "noise free" (i.e. the Y_i are actually contained in the subspaces defined by A_i and the mapping $\mathcal{A}_i(\cdot) = A^T(\cdot)A$), then we get the simple expression:

$$\sum_{i=1}^{q} A_i \hat{X} A_i = \sum_{i=1}^{q} Y_i$$

3-C Projection Space Expressions

Here we provide expressions for the $n = 3$, $m = 1$ projection space case corresponding to reconstruction of 3×3 matrices from 1-dimensional observations. We will also show how to apply the current results to the case of any set of 1-dimensional projections confined to a plane (2-dimensional subspace).

For the case under consideration $A(\theta, \phi)$ is a unit 3-vector, thus A is of the form:

$$A(\theta, \phi) = \begin{bmatrix} \sin(\theta)\cos(\phi) \\ \sin(\theta)\sin(\phi) \\ \cos(\theta) \end{bmatrix}$$

where we have chosen a longitude/co-latitude parameterization of the unit sphere, as shown in Figure 3-21. For a given $X = [X_{ij}]$ the resulting observations $Y(\theta, \phi)$ are then of the form:

$$Y(\theta, \phi) = [c_1 + c_2 \cos(2\phi) + c_3 \sin(2\phi)] + [c_4 - c_2 \cos(2\phi) - c_3 \sin(2\phi)] \sin(2\theta)$$

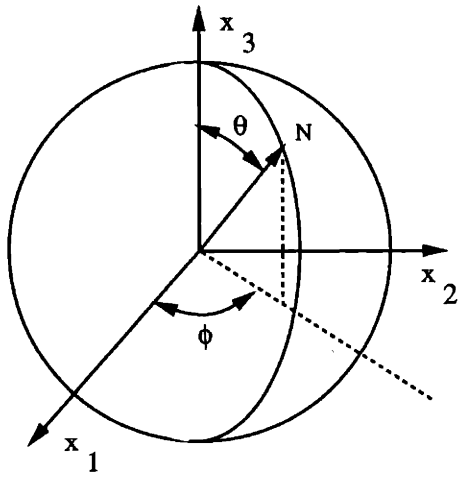

Figure 3-21: Parameterization of unit sphere.

$$
\begin{aligned}
& + [c_5 \sin(\phi) + c_6 \cos(\phi)] \cos(2\theta) \\
= \; & [c_1 + c_4 \cos(2\theta)] + c_6 \sin(2\theta) \cos(\phi) + c_5 \sin(2\theta) \sin(\phi) \\
& + [c_2(1 - \cos(2\theta))] \cos(2\phi) + [c_3(1 - \cos(2\theta))] \sin(2\phi).
\end{aligned}
$$

We have shown two factorizations of the result to show that the function is a harmonic surface with one harmonic in θ and two harmonics in ϕ. The coefficients are given by:

$$
\begin{aligned}
c_1 &= (X_{11} + X_{22} + 2X_{33})/4 \\
c_2 &= (X_{11} - X_{22})/4 \\
c_3 &= X_{12}/2 \\
c_4 &= (-X_{11} - X_{22} + 2X_{33})/4 \\
c_5 &= X_{23} \\
c_6 &= X_{13}.
\end{aligned}
$$

Planar Projections

We show here that we may generalize the results for the $n = 2$ case in a different way. The 1-dimensional projections considered above had no constraints on them. If we constrain our 1-dimensional projections to lie in a plane (a 2-dimensional subspace) then the situation for any n (matrix size) is equivalent to the $n = 2$ case. Note that we cannot completely recover the matrix X from such information (though we can

recover its projection onto the plane), but the equivalence is interesting.

To see the equivalence, suppose the 2 columns of V span the plane of interest. Then any projection matrix A in this plane may be written as

$$A(\theta) = V \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$$

for some θ. The corresponding projection observation $Y(\theta)$ is then given by

$$Y(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \end{bmatrix} V^T X V \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \end{bmatrix} \tilde{X} \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$$

where $\tilde{X} = V^T X V$ is 2×2. But this is the same as the $n = 2$ case.

3-D Proof of Result 20

We will use the projection space formalism to show that for small enough perturbations the LLSE solution of (3.7) will still remain positive definite. First we need the following result:

Lemma 6 (LS Sensitivity [61]) *Suppose x^* and $x^* + \delta x$ satisfy:*

$$\begin{aligned} x^* &= \arg\min \|Ax - b\|_2^2 \\ x^* + \delta x &= \arg\min \|Ax - b + \delta b\|_2^2 \end{aligned}$$

where A is $m \times n$, $m \geq n$ with full rank, and δb is an m-vector and δx is an n-vector.

Then the following is true:

$$\|\delta x\|_2 \leq \sigma_{\min}^{-1}(A) \|\delta y\|_2$$

where $\sigma_{\min}(A)$ is the minimum singular value of A.

Proof If $x^* + \delta x$ is the minimum norm solution of $\min \|Ax - b + \delta b\|_2$ then it must satisfy:

$$x^* + \delta x = A^+(b + \delta b)$$

where A^+ is the Moore-Penrose psudeo-inverse of A. But since x^* is the solution of the unperturbed problem $\min\|Ax - b\|_2$ we know that $x^* = A^+b$. Thus we have

$$\delta x = A^+ \delta b$$

Using the definition of the psuedo-inverse in [61, pg. 243] and the definition of the matrix 2-norm we obtain:

$$\begin{aligned} \|\delta x\|_2 &\leq \|A^+\|_2 \|\delta y\|_2 \\ &= \sigma_{\max}(A^+)\|\delta y\|_2 \\ &= \sigma_{\min}^{-1}(A)\|\delta y\|_2 \end{aligned}$$

and the result is shown. □

Now, by hypothesis, in the symmetric vector space S, we have that:

$$\begin{aligned} x^* &= \arg\min\|\mathbf{y}^* - \mathbf{A}x\|_2^2 \\ x^* + \delta x &= \arg\min\|\mathbf{y}^* + \delta\mathbf{y} - \mathbf{A}x\|_2^2 \end{aligned}$$

where x^* is the representation of X^* in S, $\mathbf{A} = [\tilde{A}_1^T|\tilde{A}_2^T|\ldots]^T$ is the matrix composed of the equivalent vector space images of the A_i under the map Γ, and $\mathbf{y}^* = [y_1^{*T}|y_2^{*T}|\ldots]^T$ is the vector composed of the representations of the Y_i^*, and similar definitions hold for the perturbed vectors $\mathbf{y}^* + \delta\mathbf{y}$ and $x^* + \delta x$. Now since the A_i are full rank and the solutions are the unique minimum norm solutions we know that $\mathbf{A}$ is full rank. Thus we may apply Lemma 6 to obtain:

$$\|\delta x\|_2 \leq \sigma_{\min}^{-1}(\mathbf{A})\|\delta\mathbf{y}\|_2 \tag{3.28}$$

But using the definitions of $\mathbf{y}$ and the relationship between the original and symmetric space norms we obtain for the last term:

$$\begin{aligned} \|\delta\mathbf{y}\|_2^2 &= \sum_{i=1}^{q} \|\delta y_i\|_2^2 \\ &= \sum_{i=1}^{q} \|\delta Y_i\|_F^2 \end{aligned}$$

where δy_i is the representation of δY_i with respect to a symmetric basis. Thus

$$\|\delta \mathbf{y}\|_2 = \sqrt{\sum_{i=1}^{q} \|\delta Y_i\|_F^2}$$

Similarly

$$\begin{aligned} \|\delta x\|_2 &= \|\delta X\|_F \\ &\geq \|\delta X\|_2 = \sigma_{\max}(\delta X) \end{aligned}$$

Combining these with (3.28) yields the desired result:

$$\sigma_{\max}(\delta X) \leq \sqrt{\sum_{i=1}^{q} \|\delta Y_i\|_F^2}$$

The second part of the result follows from the fact that given two matrices M, N with M non-singular, if $\sigma_{\min}(M) \geq \sigma_{\max}(N)$ then $M + N$ will be non-singular. Also for a symmetric matrix X with nonnegative eigenvalues $\sigma(X) = \lambda(X)$.

3-E Classification of PSD Approximations

We present a summary in Figure 3-22 of the different approximations to the PSD cone we have presented. The name of the approximation refers to the subsection on approximating the PSD cone that discussed that particular approximation. The type refers to one of the following two approximating types discussed in Section 3.3.2:

1. Linear inequality constraint on x:

$$Ex \geq f$$

2. Linear inequality constraint on a linearly transformed variable $\bar{x}$:

$$\begin{aligned} x &= G\bar{x} \\ E\bar{x} &\geq f \end{aligned}$$

As it is difficult to write the general formulas for the case of coordinate boxes, we have shown the $n = 3$ case only. Recall that $n_i = \Xi(N_i)$ with N_i a support hy-

Name	Type (Int/Ext)	E	f	G
Finite Support	1 (Ext)	$[n_1\|n_2\|\cdots\|n_q]^T$	0	—
Convex Polyhedron	2 (Int)	$\begin{bmatrix} I \\ \hline 1 & 1 & \cdots & 1 \\ -1 & -1 & \cdots & -1 \end{bmatrix}$	$\begin{bmatrix} \bigcirc \\ \hline 1 \\ 1 \end{bmatrix}$	$[\nu_1\|\nu_2\|\cdots\|\nu_q]$
Coordinate Box (n=3 case)	1 (Int)	$\begin{bmatrix} I \\ \hline 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} \underline{X}_{11} \\ \sqrt{2}\underline{X}_{12} \\ \sqrt{2}\underline{X}_{13} \\ \underline{X}_{22} \\ \sqrt{2}\underline{X}_{23} \\ \underline{X}_{33} \\ \hline -\sqrt{2}\overline{X}_{12} \\ -\sqrt{2}\overline{X}_{13} \\ -\sqrt{2}\overline{X}_{23} \end{bmatrix}$	—
Extreme Ray	2 (Int)	I	0	$[r_1\|r_2\|\cdots\|r_q]$
Geršgorin Box	1 (Int)	$[ng_1\|ng_2\|\cdots\|ng_q]^T$	0	—
Interior Exterior	1	$[n_1(t)\|n_2(t)\|\cdots\|n_q(t)]^T$	0	—

Figure 3-22: Summary of different non-iterative approximations to PSD_n.

perplane normal, $\nu_i = \Xi(V_i)$, with V_i a polhedron vertex, $r_i = \Xi(R_i)$, with R_i an extreme ray, $ng_i = \Xi(Ng_i)$, with Ng_i a Geršgorin normal, and $n_i(t) = \Xi(N_i(t))$, with $N_i(t)$ the perturbed normal $N(0) - \alpha(t)I$. The scalars $\overline{X}_{ij}$ and $\underline{X}_{ij}$ are the entry bounds of the *entry* interval matrix set. More detail may be found under the separate approximations.

3-F Proof of Result 21

We start with an observation. When **A** is of full rank, the solutions of Problem 3 and Problem 4 for any convex approximating set are unique. These properties follow from convexity of the cost function (equivalent to the Euclidean norm), the approximating sets, and the PSD set, see [63, Prop. A.35(g)].

We consider two cases. First, if Algorithm 1 produces a PSD matrix at some iteration then we are done, since, as argued previously, any PSD solution must be the

the unique LLSE solution to Problem 3. Also, if a PSD matrix is produced at any iteration, (including the first) it is a fixed point of the algorithm (since no additional constraints will be added to the approximating set $\widetilde{\mathrm{PSD}}_n(k)$ set in this case).

Now suppose a PSD matrix is not produced at some stage of the algorithm. At any stage of the algorithm with indefinite iterate $X(k)$ we add a constraint that prohibits this solution at the next stage of the algorithm. Since this solution was the *unique* minimum cost solution at that stage and since the set of possible solutions is *strictly* smaller at each stage, it is clear that the cost must *strictly increase* at each stage that there is a non-zero change in the estimate (i.e. at each stage that $X(k)$ is indefinite). Futher this cost is bounded above by the minimum cost corresponding to the solution of Problem 3. Thus the cost and the sequence $X(k)$ must converge (since the change in the iterates must converge to zero). Now from the structure of the algorithm, the updates converge to zero if and only if the iterates converge to a PSD matrix. But as we have argued, a PSD solution to Problem 4 with an external approximating set must be the unique solution to Problem 3.

3-G Evaluation of Gradients and Other Quantities

3-G.1 Gradient of $C(R)$

Here we find the gradient $\nabla C(R)$ of $C(R)$. Recall, the gradient is defined as $\nabla C(R) = \partial C(R)/\partial R$. Now

$$\begin{aligned} C(R) &= \sum_{i=1}^{q} \left\| Y_i - A_i^T R^T R A_i \right\|_F^2 \\ &= \sum_{i=1}^{q} \mathsf{Tr}\left[(Y_i - A_i^T R^T R A_i)(Y_i - A_i^T R^T R A_i) \right] \\ &= \sum_{i=1}^{q} \mathsf{Tr}\left(Y_i^2 - A_i^T R^T R A_i Y_i - Y_i A_i^T R^T R A_i + A_i^T R^T R A_i A_i^T R^T R A_i \right). \end{aligned}$$

Using the properties of matrix differentiation [21] and the trace we obtain for a general R:

$$\frac{\partial C(R)}{\partial R} = \sum_{i=1}^{q} -2(A_i^T R^T)^T (A_i Y_i)^T - 2(Y_i A_i^T R^T)^T A_i^T + 4 R A_i A_i^T R^T R A_i A_i^T$$

$$= -4R\sum_{i=1}^{q} A_i \left(Y_i - A_i^T R^T R A_i\right) A_i^T.$$

Finally, if we were to restrict R to be a symmetric square root, the effect of the off diagonal terms of R on the resulting gradient would have to be doubled, since they appear twice in R. As a result, the gradient $\nabla C(R_{\text{symm}})$ for the case of a symmetric R is given by:

$$\nabla C(R_{\text{symm}}) = \nabla C^T(R_{\text{gen}}) + \nabla C(R_{\text{gen}}) - \textbf{diag}\nabla C(R_{\text{gen}})$$

where $\nabla C(R_{\text{gen}})$ is the expression for the gradient of a general R given above and $\textbf{diag}M$ is a matrix composed of the diagonal of M. Note that $\nabla C(R_{\text{symm}})$ is symmetric so the symmetry of R is maintained in a gradient descent algorithm.

3-G.2 Proof of Result 22

Our result will follow from Theorem 2.1.6 of [66] if we can show that $C(R)$ possesses the following the following properties:

(i) $C(R) \geq 0$ for every R.

(ii) $C(R)$ is continuously differentiable and the level sets

$$\{R \,|\, C(R) \leq \alpha\}$$

are bounded for any finite α.

Now $C(R)$ is always positive and is continuously differentiable so all we need to show is the boundedness of the level sets.

To show property (ii), consider the level set $C(R) \leq K$. Recall

$$C(R) = \sum_{i=1}^{q} \left\| Y_i - A_i^T X A_i \right\|_F^2 = \|\mathbf{A}x(R) - \mathbf{y}\|_2^2$$

where the second form is the cost in the symmetric vector space representation and $x(R)$ is the representation of $X = R^T R$. Now

$$\|\mathbf{A}x(R) - \mathbf{y}\|_2 \geq \|\mathbf{A}x(R)\|_2 - \|\mathbf{y}\|_2.$$

Thus

$$\begin{aligned} \|\mathbf{A}x(R) - \mathbf{y}\|_2 + \|\mathbf{y}\|_2 &= \sqrt{C(R)} + \|\mathbf{y}\|_2 \\ &\geq \|\mathbf{A}x(R)\|_2 \\ &\geq \sigma_{\min}(\mathbf{A})\|x(R)\|_2 \end{aligned}$$

where $\sigma_{\min}(\mathbf{A})$ is the minimum singular value of $\mathbf{A}$. Since $\|\mathbf{y}\|_2$, the observation, is bounded by assumption, $C(R) \leq \alpha$, and $\sigma_{\min}(\mathbf{A})$ is nonzero (because $\mathbf{A}$ is of full rank), we have that $\|x(R)\|_2$ and hence $\|X(R)\|_F = \|R^T R\|_F$ is bounded. Since $\|R^T R\|_F \geq \|R^T R\|_2 = \|R\|_2^2$ we have shown that R is bounded.

3-G.3 Finding γ to Minimize $C(R(k) - \gamma \nabla C(R(k)))$

In this section we find γ to minimize $C(R(k) - \gamma \nabla C(R(k)))$, yielding a variable step-size rule in Algorithm 2 that often yields the fastest convergence. For notational convenience we will suppress all reference to k and henceforth assume that all quantities are evaluated at iteration k. We will also supress the dependence of $C(R)$ and $\nabla C(R)$ on R where no confusion will occur. Now

$$\begin{aligned} C(R - \gamma \nabla C) &= \sum_{i=1}^{q} \left\| Y_i - A_i^T \left[B - \gamma \nabla C \right]^T \left[B - \gamma \nabla C \right] A_i \right\|_F^2 \\ &= \sum_{i=1}^{q} \left\| E_i + \gamma F_i + \gamma^2 G_i \right\|_F^2 \end{aligned}$$

where the coefficient matrices are defined as follows:

$$\begin{aligned} E_i &= Y_i - A_i^T R^T R A_i \\ F_i &= A_i^T \left(\nabla C^T R + R^T \nabla C \right) A_i \\ G_i &= -A_i^T \nabla C^T \nabla C A_i. \end{aligned}$$

Taking the partial derivative of C with respect to γ and setting it equal to 0 we obtain:

$$\begin{aligned} &\partial C(R - \gamma \nabla C)/\partial \gamma = \\ &\quad \sum_{i=1}^{q} 2\langle E_i, F_i \rangle + 2\gamma \left(\|F_i\|_F^2 + 2\langle E_i, G_i \rangle \right) + 6\gamma^2 \langle F_i, G_i \rangle + 4\gamma^3 \|G_i\|_F^2 = 0. \end{aligned}$$

Thus we seek the roots of the cubic equation

$$p_0 + p_1\gamma + p_2\gamma^2 + p_3\gamma^3 = 0 \tag{3.29}$$

where

$$\begin{aligned} p_0 &= \sum_{i=1}^{q} \langle E_i, F_i \rangle \\ p_1 &= \sum_{i=1}^{q} \left(\|F_i\|_F^2 + 2\langle E_i, G_i \rangle \right) \\ p_2 &= \sum_{i=1}^{q} 3\langle F_i, G_i \rangle \\ p_3 &= \sum_{i=1}^{q} 2\|G_i\|_F^2. \end{aligned}$$

We shall now show how to evaluate the coefficients p_i by working in the symmetric space S using the isomorphism Ξ. As before, define $\mathbf{A} = [\tilde{A}_1^T | \tilde{A}_2^T | \cdots | \tilde{A}_q^T]^T$, with $\tilde{A}_i = \Gamma(A_i)$, $\mathbf{y} = [y_1^T | y_2^T | \cdots | y_q^T]$, with $y_i = \Xi(Y_i)$, and $x = \Xi(B^T B)$ (the current estimate). Now define the new vectors $r = \Xi(\nabla C^T R + R^T \nabla C)$, as the representation of the symmetric matrix $\nabla C^T R + R^T \nabla C$, and $s = \Xi(\nabla C^T \nabla C)$ as the representation of the symmetric matrix $\nabla C^T \nabla C$. With these quantities the coefficients p_i are given by:

$$\begin{aligned} p_0 &= r^T \mathbf{A}^T (\mathbf{y} - \mathbf{A}x) \\ p_1 &= \|\mathbf{A}r\|_2^2 - 2s^T \mathbf{A}^T (\mathbf{y} - \mathbf{A}x) \\ p_2 &= -3r^T \mathbf{A}^T \mathbf{A} s \\ p_3 &= 2\|\mathbf{A}s\|_2^2 \end{aligned}$$

In summary, to find the minimizing γ, the cubic (3.29) must be solved for its minimum root at each iteration of the algorithm. The coefficients of the cubic are given by the p_i above. Note that the quantites $\mathbf{A}$ and $\mathbf{y}$ do not change from iteration to iteration and that the quantity $\mathbf{y} - \mathbf{A}x$ is the residual at the current iteration.

3-G.4 Gradient of $C(X)$

Here we find the gradient $\nabla C(X)$ of $C(X)$. Now

$$\begin{aligned} C(X) &= \sum_{i=1}^{q} \left\| Y_i - A_i^T X A_i \right\|_F^2 \\ &= \sum_{i=1}^{q} \mathsf{Tr}\left[(Y_i - A_i^T X A_i)(Y_i - A_i^T X A_i) \right] \\ &= \sum_{i=1}^{q} \mathsf{Tr}\left(Y_i^2 - A_i^T X A_i Y_i - Y_i A_i^T X A_i + A_i^T X A_i A_i^T X A_i \right). \end{aligned}$$

Using the properties of matrix differentiation and the trace we obtain:

$$\begin{aligned} \frac{\partial C(X)}{\partial X} &= \sum_{i=1}^{q} -A_i (A_i Y_i)^T - (Y_i A_i^T)^T + A_i (A_i A_i^T X A_i)^T + (A_i^T X A_i A_i^T)^T A_i^T \\ &= -2 \sum_{i=1}^{q} A_i \left(Y_i - A_i^T X A_i \right) A_i^T. \end{aligned}$$

We may easily obtain this result another way by considering the equivalent expression in the symmetric space. To this end, let $x = \Xi(X)$ be the representation of X, $\nabla c(k) = \Xi(\nabla C(k))$ be the representation of the gradient, and $\mathbf{A}$ and $\mathbf{y}$ be defined as in (3.5). With this notation $C(X) = \|\mathbf{y} - \mathbf{A}x\|_2^2$ and $\nabla c(k) = \partial C(x)/\partial x$. This is simply $\nabla c(k) = -2\mathbf{A}^T(\mathbf{y} - \mathbf{A}x)$. This expression can be seen to be equivalent to the representation of the symmetric matrix $\nabla C(k)$ provided by the first approach.

3-G.5 Proof of Result 23

To prove convergence we invoke results from [63]. The application of these results depends on $C(X)$ satisfying certain properties: we must show strong convexity of $C(X)$ and Lipschitz continuity of $\nabla C(X)$. To make this task easier we shall work in the space of symmetric matrices. Recall that:

$$C(X) = \sum_{i=1}^{q} \left\| Y_i - A_i^T X A_i \right\|_F^2 = \|\mathbf{y} - \mathbf{A}x\|_2^2$$

where x is the representation of X in S, $\tilde{\mathbf{A}} = [\tilde{A}_1^T | \tilde{A}_2^T | \cdots]^T$ is the matrix composed of the images of the A_i under the map Γ, and $\mathbf{y}^* = [y_1^T | y_2^T | \cdots]^T$ is the vector composed

of the representations of the Y_i ($y_i = \Xi(Y_i)$). From above we have that

$$\nabla C(X) = -2\sum_{i=1}^{q} A_i \left(Y_i - A_i^T X A_i\right) A_i^T$$

and in the vector space of symmetric matrices, the representation of this symmetric matrix is simply given by

$$\nabla c(x) = -2\mathbf{A}^T(\mathbf{y} - \mathbf{A}x).$$

Strong Convexity of $C(X)$

To show strong convexity of $C(X)$ we must show that there exists some $\alpha > 0$ such that

$$\langle (\nabla C(X_1) - \nabla C(X_2)), (X_1 - X_2) \rangle \geq \alpha \|X_1 - X_2\|_F^2 \qquad \forall\, X_1, X_2.$$

Equivalently, we must show that

$$(\nabla c(x_1) - \nabla c(x_2))^T (x_1 - x_2) \geq \alpha \|x_1 - x_2\|_2^2 \qquad \forall\, x_1, x_2.$$

Now

$$\begin{aligned} (\nabla c(x_1) - \nabla c(x_2))^T (x_1 - x_2) &= (x_1 - x_2)^T \mathbf{A}^T \mathbf{A} (x_1 - x_2) \\ &= \|\mathbf{A}(x_1 - x_2)\|_2^2 \\ &\geq \sigma_{\min}^2(\mathbf{A}) \|x_1 - x_2\|_2^2 \end{aligned}$$

and we have shown strong convexity with $\alpha = \sigma_{\min}^2(\mathbf{A})$.

Lipschitz Continuity of $\nabla C(X)$

To show Lipschitz continuity of $\nabla C(X)$ we must show that there exists a constant β such that

$$\|\nabla C(X_1) - \nabla C(X_2)\|_F \leq \beta \|X_1 - X_2\|_F \qquad \forall\, X_1, X_2.$$

Equivalently, we must show that

$$\|\nabla c(x_1) - \nabla c(x_2)\|_2 \leq \beta \|x_1 - x_2\|_2 \qquad \forall\, x_1, x_2.$$

Now

$$\begin{aligned} \|\nabla c(x_1) - \nabla c(x_2)\|_2 &= \|\mathbf{A}^T(\mathbf{A}x_1 - \mathbf{A}x_2)\|_2 \\ &\leq \sigma^2_{\max}(\mathbf{A})\|x_1 - x_2\|_2 \end{aligned}$$

where $\sigma_{\max}(\mathbf{A})$ is the largest singular value of $\mathbf{A}$. Thus we have shown Lipschitz continuity of $\nabla C(X)$ with $\beta = \sigma^2_{\max}$.

Convergence

With the properties of $C(X)$ we have shown above Algorithm 3, our convergence result is essentially contained in Propositions 3.4 and 3.5 of [63].

3-G.6 Finding γ to Minimize $C(X(k) - \gamma\nabla C(X(k)))$

Here we find the minimizing γ to $C(X(k) - \gamma\nabla C(X(k)))$. Computing γ this way at each iteration yields a variable step size selection rule which often yields faster convergence of the algorithm than simply using a constant stepsize. For convenience we will suppress the reference to the iteration k and functional dependence of quantities on X where it will cause no confusion. Now

$$\begin{aligned} C(X - \gamma\nabla C) &= \sum_{i=1}^{q} \left\| Y_i - A_i^T (X - \gamma\nabla C) A_i \right\|_F^2 \\ &= \|\mathbf{y} - \mathbf{A}(x - \gamma\nabla c)\|_2^2 \end{aligned}$$

where $\mathbf{A}$, $\mathbf{y}$, ∇c, and x are as defined above. Using the second expression, it is straightforward to show that the value of γ which minimizes $C(X - \gamma\nabla C)$ is given by

$$\gamma_{\min} = -\frac{\nabla c^T \mathbf{A}^T(\mathbf{y} - \mathbf{A}x)}{\|\mathbf{A}\nabla c\|_2^2}.$$

But, from above we have that $\nabla c = -2\mathbf{A}^T(\mathbf{y} - \mathbf{A}x)$, thus

$$\gamma_{\min} = \frac{1}{2}\frac{\|\nabla c\|_2^2}{\|\mathbf{A}\nabla c(k)\|_2^2}.$$

Using the definition of $\mathbf{A}$, the mapping $\Gamma(A_i)$, and the equivalence of a Frobenius norm of a symmetric matrix and the 2-norm of its representation, we obtain:

$$\gamma_{\min} = \frac{1}{2} \frac{\|\nabla C(k)\|_F^2}{\sum_{i=1}^{q} \|A_i^T \nabla C(k) A_i\|_F^2}.$$

where all these quantities are to be evaluated at iteration k.

3-H Successive Rotation

Here we discuss an iterative approach which does not appear to work very well in practice, yet is interesting in its use of the geometry of the problem. Consider that any symmetric matrix X may be decomposed as $X = U\Lambda U^T$ where U is unitary and Λ is a diagonal matrix of the eigenvalues. We can consider that in observing the symmetric matrix X we are observing the diagonal matrix Λ in a *rotated* set of coordinates, with the rotation provided by U. In our rotated set of coordinates it is difficult to tell if X is positive semi-definite or not. But if somehow we knew the proper underlying rotation, then by applying the *inverse* rotation we could easily tell if the matrix was a member of the PSD set, as the resulting diagonal would have to be non-negative. With this insight, the idea of the algorithm is this: At step k we estimate the rotation $U(k)$ corresponding the desired underlying optimal solution to Problem 3. With this rotation, we rotate coordinates, transforming the geometry as given by A_i. Then in this transformed set of coordinates, we solve Problem 4 with some approximate constraint $\widetilde{\mathrm{PSD}}_n(k)$ on the solution. Thus we solve the transformed problem:

$$D^* = \underset{D \in \widetilde{\mathrm{PSD}}_n(k)}{\arg\min} \sum_{i=1}^{q} \left\| Y_i - A_i^T U^T(k) D U(k) A_i \right\|_F^2. \tag{3.30}$$

The resulting matrix estimate is then used to update the rotation rotation estimate. Note that if the matrix U corresponds to the true underlying solution, and if $\widetilde{\mathrm{PSD}}_n = \mathrm{PSD}_n$ then the solution D^* of (3.30) should be a diagonal matrix. Formally the algorithm is given by the following

Algorithm 7 (Successive Rotation)

Step 1) *Make initial guess for the unitary (rotation) matrix* $U(0)$. *Set* $k = 0$.

Step 2) *Pick the constraint set* $\widetilde{PSD}_n(k)$.

Step 3) *Solve the transformed problem (3.30) for $D^*(k)$ using $U(k)$.*

Step 4) *Find the next estimate as $X(k+1) = U(k)D^*(k)U(k)^T$.*

Step 5) *Perform an eigenanalysis on $X(k+1)$ to obtain $U(k+1)$ as $X(k+1) = U(k+1)\Lambda U(k+1)^T$.*

Step 6) *Goto Step 2).*

We could choose a variety of constraint sets $\widetilde{\mathrm{PSD}}_n(k)$ for Step 2), including static constraints. For example, we could choose the (external) static constraint that the diagonal elements must be positive. This is easy to implement, but might allow an indefinite matrix fixed point. Another possibility is to choose one of the static approximations discussed earlier. For example, we could choose the Geršgorin polygonal cone.

The above constraint set examples have been static. To help convergence we could try varying $\widetilde{\mathrm{PSD}}_n(k)$ from step to step. One approach we have tried is to apply a congruence transformation to the matrix D, essentially warping the constraint set. That is, rather than solving (3.30) in Step 3), we would solve the transformed problem:

$$D^* = \underset{D \in \widetilde{\mathrm{PSD}}_n(k)}{\arg\min} \sum_{i=1}^{q} \left\| Y_i - A_i^T U^T(k) Q^T D Q U(k) A_i \right\|_F^2 .$$

where Q is an invertible matrix. The congruence tranformation preserves the inertia of a matrix, thus this transformation would not change the positive semi-definiteness of any solution. This transformation may be viewed equivalently as warping the constraint set. The transformation must then be "undone" in Step 4) so that $X(k+1) = U(k)Q^{-T}D^*(k)Q^{-1}U(k)^T$. The effect of this scheme seems to be inconsistent, depending on the combination of the specific Q used coupled with the particulars of the problem under consideration.

We know that for convergence we would like to use a constraint set with constraints that match the exact ones for a diagonal matrix. In other words, we desire a constraint set whose intersection with the subspace corresponding to $X_{ij} = 0$, $i \neq j$ matches that of the PSD set. If a constraint set has this property, then we can see that the true minimizing solution of Problem 3 will be a fixed point of the algorithm. The Geršgorin polygon set is one such set that has this property. Unfortunately, even with such a constraint set, while the algorithm will sometimes converge to the true minimum, it also appears to often converge to nearby local minima. Perhaps this is

due to the nonsmooth nature of the eigenvalue constraint [48]. Thus, in its present state the algorithm appears of limited practical use. The approach still appears to be an interesting use of the underlying geometry of the problem.

Chapter 4

Curvature of Surfaces and Their Shadows

4.1 Introduction

In this chapter we examine the relationship between the n-dimensional surfaces of smooth, strictly convex objects and the m-dimensional surfaces of their orthogonal projections, or shadows. Our main results concern the relationships between the local properties of the surface at a point and those of its shadow. Specifically, we show that the curvature Hessian of the projection at a boundary point is simply the projection of the curvature at the point's pre-image. This notion of projection which is appropriate for the symmetric Hessian matrix is precisely that which we defined in Chapter 2 for arbitrary symmetric matrices, and motivates our examination of that problem. Using the results from Chapter 2 we present consise necessary and sufficient conditions for solution of the inverse problem of determining the surface curvature at a point, given the curvatures of a series of projections involving the point. These local relationships are then combined with a curvature based object representation on the Gaussian sphere to both construct the projection shadows of objects and to elucidate the inverse problem of reconstructing global object shape from shadows. The results presented here serve to illuminate and extend the work of Van Hove [7] for obtaining the 2-dimensional shadow of an object in 3-dimensional space. We presented the results of this chapter in [68].

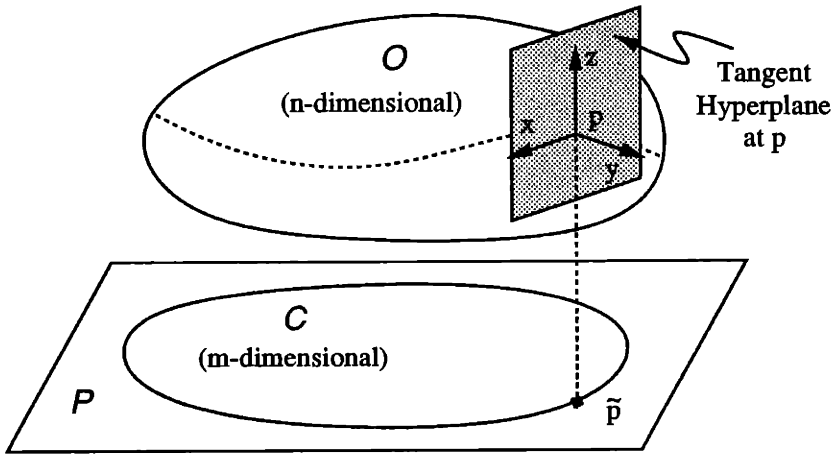

Figure 4-1: Problem definition.

Mathematical Preliminaries

We are concerned with the orthogonal projection of the n-dimensional surface $\mathcal{O}$ of an object in $(n+1)$-dimensional space onto an $(m+1)$-dimensional projection subspace $\mathcal{P}$, to obtain a shadow with an m-dimensional surface, see Figure 4-1. A guide to notation is presented in Appendix 4-A. In general, our attention is restricted to smooth, strictly convex objects (termed "rotund" [46]). We represent the curvature of the surface $\mathcal{O}$ at a point p by the symmetric Hessian matrix at the point (notation $\mathsf{H}_{\mathcal{O}}(p)$), i.e., by the matrix of second partial derivatives of the surface in some local tangent-based coordinate system [69], as illustrated below.

The boundary of the shadow in the subspace $\mathcal{P}$ will be a "curve" (actually a surface in $\mathcal{P}$), which we label $\mathcal{C}$. This lower dimensional surface, in turn, will be the shadow of a "curve," termed the contour generator (CG) [15, pp. 106], on the surface $\mathcal{O}$ of the object. The points of the CG are thus precisely those that map to the boundary $\mathcal{C}$ under projection. If p is such a point on the CG, we label its image in $\mathcal{C}$ by $\tilde{p}$. In addition, we assume the dimension of $\mathcal{P}$ is at least 2, so that the curvature of $\mathcal{C}$ is well defined

In our main result we relate $\mathsf{H}_{\mathcal{O}}(p)$ to $\mathcal{H}_{\mathcal{C}}(\tilde{p})$. Specifically, in Section 4.2 we show that $\mathsf{H}_{\mathcal{C}}(\tilde{p}) = (S^T \mathsf{H}_{\mathcal{O}}^{-1}(p) S)^{-1}$, where the columns of S form an orthonormal basis for a subspace defined by $\mathcal{P}$ and the tangent space of $\mathcal{O}$ at p. This result serves to generalize and unify existing work, such as that in [7]. It will also form a tie between the general symmetric matrix projection and estimation problem we examined in Chapter 2 and the following set of inverse problems. Specifically, after the treatment of the forward or

projection problem, we will examine the inverse problem of *determining* the curvature of a surface at a point from a series of projections involving the point. Necessary and sufficient conditions are presented for recovering $\mathsf{H}_{\mathcal{O}}^{-1}(p)$ from a series of observations $\mathcal{H}_{C_i}^{-1}(\tilde{p})$, $1 \leq i \leq q$. These conditions will make use of our results in Chapter 2.

In Section 4.3 we combine these local results with a curvature based representation to make a series of global statements. A formalism based on the work in [7] will be given, allowing us to conveniently find the projection of an object onto an arbitrary subspace. These relationships will serve to generalize and clarify the work in [7, 19] on silhouette determination in two-dimensions. We subsequently address issues arising in the inverse problem of overall recovery of object shape from shadows.

In Section 4.4 we pose some questions and raise some issues for future research. We give a brief discussion there of an alternative curvature representation scheme based on the Gaussian curvature rather than the Hessian.

4.2 Local Results

We show here that the curvature of the projection at a boundary point $\tilde{p}$ of the shadow is precisely the projection of the curvature at the corresponding point p on the contour generator of the object. We follow this local projection result with necessary and sufficient conditions for the solution of the inverse problem of determining the curvature of a surface at a point from a series of projections.

4.2.1 Projection of a Surface

The situation under consideration is depicted in Figure 4-2. Pick a coordinate frame in the tangent hyperplane $\mathcal{T}$ at p and label the n coordinate directions by $\hat{t}_i$, $i = 1, \ldots, n$. Complete this local frame by appending the local outward normal direction $N(p)$ as the $(n+1)$-st coordinate direction, $\hat{y}$. Let $(t_1, t_2, \ldots, t_n, y)$ be the scalar coordinates of points with respect to this frame. Locally the surface is then representable as (the Monge parameterization [23]) $y = F(t_1, \ldots, t_n)$. The curvature is taken to be the symmetric Hessian matrix $\mathsf{H}_{\mathcal{O}}(p)$ of second partial derivatives of $F(t)$ with respect to these coordinates, i.e. the matrix whose ij-th entry is $\partial^2 y / \partial t_i \partial t_j$.

For simplicity, we translate the projection subspace $\mathcal{P}$ parallel to itself to the point of interest p, as illustrated. Denote by $\mathcal{S}$ the m-dimensional intersection of $\mathcal{T}$ and the translated $\mathcal{P}$, and let the m unit vectors $\hat{S}_i$ define a coordinate frame for this

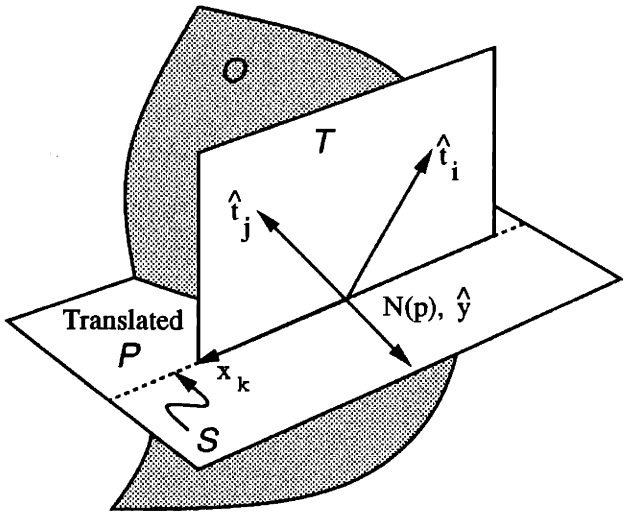

Figure 4-2: Local situation

subspace of $\mathcal{T}$. Now, the $m+1$ vectors $\{\hat{S}_i, N(p)\}$ define a local coordinate frame in the *projection* of $\mathcal{O}$ at $\tilde{p}$. As above, let $(x_1, x_2, \ldots, x_m, y)$ be the scalar coordinates of points in the projection with respect to this frame. In the projection then, the curve $\mathcal{C}$ is locally representable as $y = f(x_1, \ldots, x_m)$. The curvature of the shadow $\mathsf{H}_{\mathcal{C}}(\tilde{p})$ is the Hessian of $f(x)$ with respect to these coordinates. Finally, denote by S the $n \times m$ matrix whose columns are the representations of the vectors $\{\hat{S}_i\}$ with respect to the coordinate frame $\{\hat{t}_1, \ldots, \hat{t}_n\}$ we have defined in $\mathcal{T}$ at p. The columns of S are thus orthonormal. With this notation the main projection result is:

Result 24 (Curvature of Projection = Projection of Curvature)
The curvature Hessian of the orthogonal projection of a rotund surface at a boundary point is precisely the orthogonal projection of the curvature Hessian of the surface at the pre-image of this point: $\mathsf{H}_{\mathcal{C}}(\tilde{p}) = (S^T\, \mathsf{H}_{\mathcal{O}}^{-1}(p)\, S)^{-1}$.

The proof of Result 24 is given as Appendix 4-B. Our formulation and proof are done in a general setting, valid for any dimension, and thus represent an extension to the work in [7]. We believe they are also considerably more transparent.

Result 24 is a local one, involving only the surface properties at a point. It establishes a relationship between the second order term of the Taylor series approximation of the surface at p and the corresponding term for the projection. This simple relationship will be of use in Section 4.3 to find the complete shadow of an object. Note that this relationship is identical to (2.3) if we take $A = S$, $Y = \mathsf{H}_{\mathcal{C}}(\tilde{p})^{-1}$ and

$X = \mathsf{H}_{\mathcal{O}}^{-1}(p)$. Further, since the columns of S as we have defined it are orthonormal, the orthogonal projection of the surface induces what we have defined to be an orthogonal projection of the symmetric matrix $\mathsf{H}_{\mathcal{O}}^{-1}(p)$. This was one of our motivations in defining the mapping Γ as we did in Chapter 2.

4.2.2 The Inverse Problem – Curvature from Projections

The inverse problem of finding the curvature of a surface at a point from orthogonal projections of the surface at the point will now be examined. From Result 24 and our discussion following it, we see that the inverse problem may be viewed as the determination of the $n \times n$ symmetric matrix $\mathsf{H}_{\mathcal{O}}^{-1}(p)$ from a series of observations of curvature of the form $\mathsf{H}_{\mathcal{C}_i}^{-1}(\tilde{p}) = S_i^T \, \mathsf{H}_{\mathcal{O}}^{-1}(p) \, S_i$, where $\mathsf{H}_{\mathcal{C}_i}^{-1}(\tilde{p})$ is taken as the i-th observation and S_i is a known matrix whose orthonormal columns define the i-th subspace of projection, as described above. Using the methods of Chapter 2 we formulate this task as a standard linear estimation problem and present a simple necessary and sufficient condition for its solvability in terms of a rank test on a matrix derived from the set of S_i. Further, we bring out certain implications of this condition directly in terms of the projection subspaces $\mathcal{P}_i$.

For convenience in what follows, we take our observations to be $G_i \equiv S_i \, \mathsf{H}_{\mathcal{C}_i}^{-1}(\tilde{p}) \, S_i^T$ rather than $\mathsf{H}_{\mathcal{C}_i}^{-1}(\tilde{p})$. The latter can always be recovered from the former, since each S_i has full column rank. Again for simplicity, we let G denote $\mathsf{H}_{\mathcal{O}}^{-1}(p)$, the inverse of the Hessian of interest. Note that for strictly convex objects, the above inverses are certain to exist. With this notation the inverse problem may be phrased as follows.

Problem 7 (Local Curvature from Projections) *Determine the positive definite, symmetric, $n \times n$ matrix G, given q quadratic-form observations of the type:*

$$G_i \equiv P_i \, G \, P_i \qquad\qquad 1 \leq i \leq q \tag{4.1}$$

where the P_i are orthogonal projection matrices, i.e. are symmetric and satisfy $P_i^2 = P_i$.

In terms of the matrices $\{S_i\}$, we have $P_i \equiv S_i \, S_i^T$. Since the columns of the known $n \times m_i$ matrix S_i form an orthonormal basis for the subspace $\mathcal{S}_i$, P_i is an orthogonal projector (see e.g. [37] for a definition) onto this subspace and thus defines projection i. This means our observations G_i are invariant to the particular bases S_i chosen for the projection subspaces $\mathcal{S}_i$.

This problem is precisely of the form treated in Chapter 2, particularly Section 3.3. This problem will also appear in our consideration of the problem of reconstructing an ellipsoid from its orthogonal projections as Problem 8. In particular we may use our knowledge developed in Chapter 2 here. For example, we know by using the Kronecker product $\otimes$ and vec$(\cdot)$ operator defined there, that the problem is linear in the elements of the target matrix G. This fact can be demonstrated by writing (4.1) as:

$$\text{vec}(G_i) = (P_i \otimes P_i)\,\text{vec}(G) \tag{4.2}$$

As discussed in Chapter 2, the difficulty with this formulation is that the symmetry of the observations G_i and the target G is lost. While the vector vec(G) has n^2 elements, only $n(n+1)/2$ of them are independent. We overcome this difficulty as in Chapter 2 by imbedding the problem in a natural way in the space of symmetric matrices. In this space, the target matrix G is an unknown vector and an observation becomes the orthogonal projection of this vector onto a certain subspace.

Representation in Space of Symmetric Matrices

Recall that the set of $n \times n$ symmetric matrices together with the inner product $\langle A, B\rangle \equiv \text{tr}(A^T B)$ defines an $n(n+1)/2$-dimensional Euclidean space, notation S (where we have dropped the explicit mention of its dimension for convience). Recall also that this inner product induces the Frobenius norm on a matrix $\langle A, A\rangle^{1/2} = \|A\|_F$. Following Chapter 2, let $\{\, M_\ell \,|\, 1 \le \ell \le n(n+1)/2\}$ be an orthonormal basis for the space S. Then both G and G_i may be represented as vectors with respect to this basis, which we denote by γ and g_i respectively. By $\mathbf{P}_i$ denote the matrix relating γ and g_i. From Chapter 2 we know that $\mathbf{P}_i$ is the image of P_i under the map Γ defined there. With these definitions, we may represent our original relation (4.1) equivalently as:

$$g_i = \mathbf{P}_i\,\gamma \tag{4.3}$$

where it is straightforward to show that (see Chapter 2 for more detail):

$$\begin{aligned} \gamma_k &= \langle G, M_k\rangle \\ g_{i_k} &= \langle\, G_i, M_k\,\rangle \\ \mathbf{P}_{i_{jk}} &= \langle\, M_j, P_i\, M_k\, P_i\,\rangle \end{aligned}$$

for $1 \le i \le q, 1 \le j, k \le n(n+1)/2$.

From the properties of the map Γ we know that since P_i is a projector then $\mathbf{P}_i$ is an also an orthogonal projector. Further, we can see that this projection is onto the subspace of S spanned by the matrices $P_i M_j P_i$. Thus in the space S, the i-th observation is the vector g_i obtained as the projection of the (unknown) vector γ onto a subspace specified by $\mathbf{P}_i$ (or equivalently S_i).

We may now stack up all the observations (4.3) into a single vector to obtain the following overall relation:

$$\left[\begin{array}{c} g_1 \\ \hline g_2 \\ \hline \vdots \\ \hline g_q \end{array}\right] = \left[\begin{array}{c} \mathbf{P}_1 \\ \hline \mathbf{P}_2 \\ \hline \vdots \\ \hline \mathbf{P}_q \end{array}\right] \gamma \tag{4.4}$$

so

$$\mathbf{g} = \mathbf{P}\,\gamma \tag{4.5}$$

where $\mathbf{g}$ and $\mathbf{P}$ are defined in the obvious way. With this formulation, Problem 7 becomes: find the unknown vector γ (representating the surface curvature) given the (projected curvature) observations in $\mathbf{g}$ and the observation geometry specified by $\mathbf{P}$. We have thus phrased the problem as one in standard linear estimation.

Necessary and Sufficient Conditions for Solvability

As in Chapter 2, the formulation in (4.5) immediately allows us to characterize the solutions of Problem 7. Specifically, a unique solution to the problem exists if and only if the null space of the mapping in (4.5) is empty. This, in turn, is true if and only if the matrix $\mathbf{P}$ has full column rank ($= n(n+1)/2$). Note that an overdetermined set will be consistent since we have assumed no noise in our measurements. We phrase this condition formally:

Result 25 (General Condition for Solvability) *Problem 7 has a unique solution if and only if the matrix* $\mathbf{P}$ *defined in Equation 4.5 has rank equal to* $n(n+1)/2$ *(i.e. full column rank). This solution, if it exists, is given by*

$$\gamma = \mathbf{P}^L \mathbf{g} \tag{4.6}$$

where $\mathbf{P}^L$ *is any left inverse of* $\mathbf{P}$, $\mathbf{g}$ *is the vector of observations, and* γ *is the representation of* G *with respect to the basis* $\{M_\ell\}$.

This result provides us with a way to test if a given series of projections, defined by P_i (and thus S_i), is sufficient to determine G, or if others are needed. Later we present some conditions phrased *directly* in terms of the subspaces $\mathcal{P}_i$. Note that in the noise free case, the properties of the matrices $\mathbf{P}_i$ imply that $\gamma = (\sum_{i=1}^{q} \mathbf{P}_i)^{-1} (\sum_{i=1}^{q} g_i)$.

In the noisy case, we may easily find the linear least squared error (LLSE) solution to an inconsistent set of equations given in the form (4.5). If we choose $\mathbf{P}^L = \mathbf{P}^+$, the Moore-Penrose inverse of $\mathbf{P}$, the equation (4.6) will yield the LLSE solution to Problem 7 (without the definiteness constraint). If the inconsistencies are so great that the LLSE solution does not produce a solution corresponding to a positive semi-definite matrix then we may use one of the approaches from Chapter 2 to find the least squares solution subject to (4.5) and subject to a positive semi-definite constraint. Note that we may also implement these least squares solutions recursively (via recursive linear least squares), updating our current estimate of the vector γ (and thus of the matrix G) as more observations become available. In this vein, we might imagine using a recursive formulation to track the changing curvature of a *dynamically evolving* object.

Projection Space Conditions

In this section, some conditions for the reconstruction of G will be stated *directly* in terms of the projection subspaces $\mathcal{P}_i$. These conditions are corollaries of Result 25 but, being phrased in terms of the subspaces of projection, do not require computation of the matrix $\mathbf{P}$. We give conditions for projections onto combinations of hyperplanes and 2-dimensional subspaces (true planes, which are the smallest projection spaces that yield shadows with well defined curvature).

In [7, page 185] the result is given that for objects in three-dimensions ($n = 2$), a minimum of three projections onto planes is required to recover the curvature at a point. We show here that a generalization of this result to higher dimensions is that projections onto two *hyper*planes and onto a single additional 2-dimensional plane are required to find the curvature at a point. The result is:

Corollary 4 (Hyperplanes + 1) *Projection onto at least two hyperplanes and a single 2-dimensional subspace (plane) is necessary to uniquely recover G. These projections will be sufficient provided the 2-dimensional subspace is not contained in either hyperplane.*

This group constitutes a minimal set of observations to recover G, in the sense that any

other non-trivial set will increase either the number of observations or the dimension of the observations. We outline the proof in Appendix 4-C.

Now consider the case where the projection subspaces $\mathcal{P}_i$ are restricted to be hyperplanes. Applying Corollary 4 to this case, where the 2-dimensional subspace referred to there is contained in a third hyperplane, yields the following:

Corollary 5 (Hyperplanes) *If the projection subspaces $\mathcal{P}_i$ are restricted to be hyperplanes, then at least three such projections are necessary to uniquely recover G, and three will be sufficient provided the hyperplanes are distinct.*

Other statements of this kind, combining Result 25 with different, specific combinations of projection subspace dimensions, are of course possible.

4.3 Global Statements

The local curvature relationships of Result 24 and Result 25, presented in the previous section, will now be combined with a curvature based representation of objects on an enhanced Gaussian sphere to make global statements about shadow determination and shape reconstruction. From the description in the introduction, it can be seen that finding the contour generator, and hence the projection of an object is, in general, a difficult task. In what follows, a formalism will be given that allows the projection of an object onto an arbitrary subspace to be found in a convenient fashion. This formalism serves to generalize and clarify the work in [7, 19] on 2-dimensional shadow determination. Following this solution to the forward problem, we will use the enhanced Gauss representation and Result 25 to make statements about the inverse problem of object reconstruction from shadows.

4.3.1 Projection of $\mathcal{O}$ onto $\mathcal{P}$

Consider the projection of the object surface $\mathcal{O}$ onto the subspace $\mathcal{P}$ to obtain the region bounded by the curve $\mathcal{C}$. This situation is shown as link (1) in Figure 4-3. The points p of $\mathcal{O}$ can be separated into two types with respect to the projection onto $\mathcal{P}$: those that project to interior points and those that project to boundary points $\mathcal{C}$. The first type correspond to points within the object's shadow while the second type correspond to points on the object's contour generator (CG). These latter points are precisely those points p of $\mathcal{O}$ where the normal $N(p)$ is parallel to the subspace of

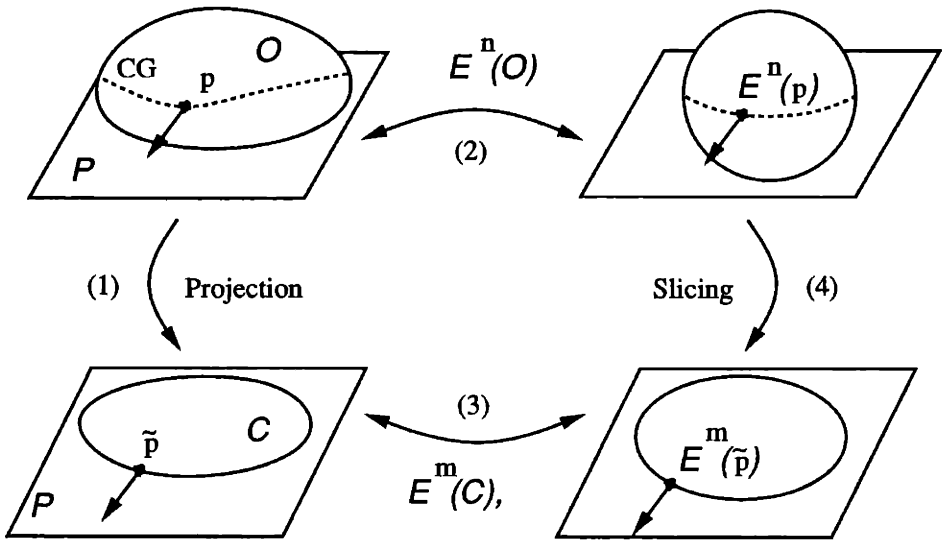

Figure 4-3: Projection and mapping relations

projection $\mathcal{P}$. The curve $\mathcal{C}$ may thus be found by first identifying the CG through the normal condition, then projecting this curve onto $\mathcal{P}$.

As can be seen, the CG and thus the curve $\mathcal{C}$ are not simple to find. In spite of this, we may show that the projection $\mathcal{C}$ inherits certain properties from the object surface $\mathcal{O}$. Specifically, if the object surface $\mathcal{O}$ is smooth and strictly convex, it is not difficult to show that $\mathcal{C}$ will also be. This provides us an easier, indirect way to find the projection.

4.3.2 The Enhanced Gauss Images $\mathcal{E}^n(\mathcal{O})$ and $\mathcal{E}^m(\mathcal{C})$

With the insights above, we consider a representation of the object better suited to the task of projection than the standard point set or functional one. Such a representation is found in an enhanced Gauss map of the object to the n-dimensional Gaussian sphere. The Gaussian sphere is a unit sphere with each point on the sphere corresponding to points of an object with the same surface normal orientation. The points corresponding to the contour generator are thus easily found on the Gaussian sphere: they lie on the great circle set obtained by the *intersection* of the Gaussian sphere and the subspace $\mathcal{P}$ [7, 70, pp. 536]. This enhanced Gauss mapping operation is shown as link (2) in Figure 4-3, and is a generalization of that presented in [19]. We define the enhanced Gauss map formally as follows:

Definition 3 (Enhanced Gauss Map – $\mathcal{E}^n(\bullet)$) *The n-dimensional enhanced Gauss map of an object $\mathcal{O}$ (notation $\mathcal{E}^n(\mathcal{O})$) is the composition of the standard Gauss map [71] with a map of each object surface point p to the pair $\{\mathsf{H}_{\mathcal{O}}(p), L(p)\}$, where $\mathsf{H}_{\mathcal{O}}(p)$ is the Hessian of the surface at p in local coordinates, and $L(p)$ is the transformation from the global coordinate system to this local coordinate system in the tangent hyperplane at p.*

It can be shown that for a rotund hypersurface $\mathcal{O}$ in $\mathbf{R}^{n+1}$ with the associated enhanced Gauss image $\mathcal{E}^n(\mathcal{O})$, the image $\mathcal{E}^n(\mathcal{O})$ (via the curvature) determines $\mathcal{O}$ up to a translation [22, 23, 24]. Thus we may equivalently work with the enhanced Gauss image of an object as with the object itself. We note that while the object is uniquely defined by its enhanced Gauss image, inverting the image to recover the object is not trivial. In principle, this inversion can always be performed, given suitable initial conditions, but it is only recently that both iterative and closed-form algorithms have been proposed for this problem [21, 72, 7].

Since the boundary $\mathcal{C}$ inherits its rotundness from $\mathcal{O}$, we may also consider an equivalent representation for $\mathcal{C}$ in terms of its m-dimensional enhanced Gauss image $\mathcal{E}^m(\mathcal{C})$. As in the case of the object surface $\mathcal{O}$, this representation determines $\mathcal{C}$ up to a translation. This tie is shown as link (3) in Figure 4-3.

4.3.3 Obtaining $\mathcal{E}^m(\mathcal{C})$ from $\mathcal{E}^n(\mathcal{O})$

In this section, a direct tie is made between $\mathcal{E}^n(\mathcal{O})$ and $\mathcal{E}^m(\mathcal{C})$, the enhanced Gauss representations for $\mathcal{O}$ and $\mathcal{C}$ respectively. The image $\mathcal{E}^m(\mathcal{C})$ depends on the *local* curvature information of $\mathcal{C}$ at each point, which in turn depends only on the local surface shape of $\mathcal{O}$ at points along the contour generator. Since our representations are curvature based, we may thus go directly from $\mathcal{E}^n(\mathcal{O})$ to $\mathcal{E}^m(\mathcal{C})$ without explicitly finding the contour generator or the curve $\mathcal{C}$. The required local relationship between the object surface curvature and the projected surface curvature at a point is provided precisely in our Result 24. By isolating this observation we have focused on the essential element of the generalization.

To obtain $\mathcal{E}^m(\mathcal{C})$ from $\mathcal{E}^n(\mathcal{O})$ we thus need only to project the Hessians of points on the contour generator onto $\mathcal{P}$. These points, as mentioned earlier, are easily found from $\mathcal{E}^n(\mathcal{O})$ as the intersection of $\mathcal{E}^n(\mathcal{O})$ and $\mathcal{P}$. All that remains is to define the new transformation to local coordinates in the projection $\tilde{L}(\tilde{p})$, which is straightforward. Pick a global coordinate frame for the *projection* contained in the subspace $\mathcal{P}$. Let

the columns of Π be the representation of these global projection axes with respect to the original global coordinate system, so the columns of Π form an orthonormal basis for $\mathcal{P}$ in global coordinates. The transformation from this set of axes in the projection to the local set at $\tilde{p}$, given by $\{\hat{S}_i, N(p)\}$ and defined in Section 4.2.1, is then given by $\tilde{L}(\tilde{p})$:

$$\tilde{L}(\tilde{p}) = \left[\begin{array}{c|c} 1 & \leftarrow 0 \rightarrow \\ \hline \uparrow & \\ 0 & S^T(p) \\ \downarrow & \end{array}\right] L(p)\ \Pi \tag{4.7}$$

where S is defined in Section 4.2.1.

We thus have the following two-step procedure to go directly from $\mathcal{E}^n(\mathcal{O})$ to $\mathcal{E}^m(\mathcal{C})$; this procedure is a generalization of the method in [7], in that it holds for any combination of object and projection dimension.

Procedure 1 ($\mathcal{E}^n(\mathcal{O})$ **to** $\mathcal{E}^m(\mathcal{C})$)

Step 1) *Identify the points of $\mathcal{O}$ on the contour generator by intersecting $\mathcal{E}^n(\mathcal{O})$ with the subspace $\mathcal{P}$.*

Step 2) *Use Result 24 to project the curvature information $\mathsf{H}_{\mathcal{O}}(p)$ onto $\mathcal{P}$ at points p along the great circle set obtained in 1). Find the new transformation $\tilde{L}(\tilde{p})$ from global to local coordinates in the projection using (4.7).*

This is a slicing operation followed by a series of local projections of curvature. Note that the method just uses knowledge of $\mathcal{E}^n(\mathcal{O})$ and $\mathcal{P}$ to obtain $\mathcal{E}^m(\mathcal{C})$, so we may work entirely in the domain of the enhanced Gauss representation. This step is shown as link (4) in Figure 4-3. This final tie completes all the links in Figure 4-3 relating the various objects and representations.

4.3.4 The Inverse Problem and the Gauss Map

We will now examine the object reconstruction problem in terms of the enhanced Gauss representation for $\mathcal{O}$. In this framework, to determine $\mathcal{O}$ (within translation) we need to determine the curvature at each surface point, as discussed in Section 4.2.2. Each projection or shadow provides information about the surface along an m-dimensional great circle set of the Gaussian sphere representation, defined by the intersection of the Gaussian sphere with the projection subspace $\mathcal{P}$. Multiple

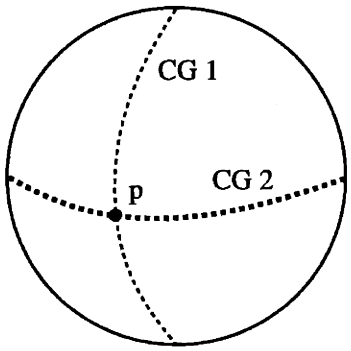

Figure 4-4: A point on two contour generators

projections of a point p correspond to places on the Gaussian sphere where these great circle sets intersect. For example, if the point p of an object in $\mathbf{R}^3$ were on the contour generator of two different projections, then it would lie in the intersection of two distinct great circles on the Gauss map of the object, as shown in Figure 4-4.

With this insight, we see that recovering the shape of an object may be viewed as a series of local curvature reconstructions at each point of the object. We view each projection as its great circle set on the Gaussian sphere. The conditions in Section 4.2.2, specifically Result 25, imply a minimum number of projections involving a given point that are required to find the local surface curvature there. This, in turn, is equivalent to a condition on the number of intersections of great circle sets at the point. Specifically, for uniform projection dimension, to determine the Hessian at p we must have enough distinct great circles intersect at the image of p on the Gaussian sphere so that $\mathbf{P}$ in (4.6) is invertible. For example, we saw that three distinct hyperplane projections at a point were necessary and sufficient for curvature recovery.

Consider the case of projection onto 3 general hyperplanes. This is the largest, non-trivial, projection dimension possible, and is thus a "best" case in that it yields the most information for each projection. From Corollary 5 and generic intersection counting arguments, it is straightforward to show that three such projections will yield two $(n-3)$-dimensional sets of points on the n-dimensional surface $\mathcal{O}$ of the object, with enough information to recover the curvature. In general, there will also be six $(n-2)$-dimensional sets of points with two hyperplanes worth of information and three $(n-1)$-dimensional sets of points with one hyperplane worth of information. For example, in $\mathbf{R}^3$ where $n=2$, this implies that projections onto three planes generically yields no points on the Gaussian sphere with three great circle intersections, six points

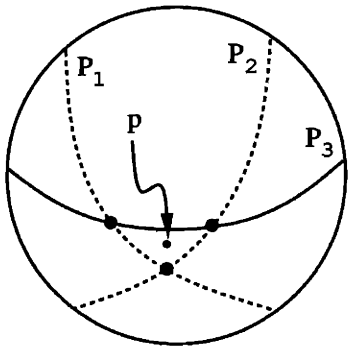

Figure 4-5: A point near the intersection of a number of projections

on the Gaussian sphere where two great circles intersect, and three 1-dimensional sets of the great circles themselves. This generic situation is depicted in Figure 4-5.

For a finite number of projections, it can thus be seen that those points with the required number of intersections will be relatively sparse. In fact, we can see that each new projection only constrains an additional finite number of $(n-3)$-dimensional sets of points on the n-dimensional surface. In general, we will have to use nearby or interpolated projection data, as for the point p in Figure 4-5. In this case, we might use the inconsistent information from nearby projections together with a least squares solution to the local problem in (4.5) to estimate a value for the point. This raises the issue of the first order relationship between the orientation of the great circle sets and the corresponding conditioning of the matrix $\mathbf{P}$. It appears desirable to have the great circle sets intersect in angles as large as possible.

4.4 Questions

We pose some questions and raise some issues for future research. One issue is raised by the insight in Section 4.3.4 above. Rather than randomly taking projections of an object, we can use the Gaussian sphere to plan a series of views that will lead to the maximum number of intersections and coverage of the sphere. What is the best way to spread the information over the whole sphere, given only a finite number of views? And how can the information from near-intersections best be used, since in general there will be many points with strictly less than the necessary number of intersections. As discussed in [7], using global restrictions on object shape, e.g. continuity, to derive the object from a *finite* set of surface curvatures gives rise to sampling questions on the Gaussian sphere and some type of surface Nyquist criterion.

In the above sections, we have represented the surface curvature by the Hessian matrix of the height function in some local coordinate system. This matrix is not the only way we may represent surface curvature. It can be shown that a rotund surface is actually determined (to within a translation) by just the determinant of the Hessian given for all normal orientations (see, for example [22, Vol. 5, pp. 304–305]). This determinant is termed the *Gaussian curvature* (notation $\mathcal{K}$) and is invariant with respect to changes in the system of local coordinates. The *scalar* function $\mathcal{K}(p)$ thus determines $\mathcal{O}$ (to a translation), and the function remains scalar regardless of the dimension of the space. This function defined on the Gaussian sphere (and hence a function of normal orientation) is called the extended Gaussian sphere and has been studied by Horn and others [20].

The use of the Gaussian curvature $\mathcal{K}$ as a representation of curvature in our enhanced Gauss mapping thus appears attractive. The key lies in the relationship between $\det[\mathsf{H}_{\mathcal{O}}(p)]$ and $\det[\mathsf{H}_{\mathcal{C}}(\tilde{p})]$, or more generally between the eigenvalues of a symmetric, definite H and those of $(S^T\mathsf{H}^{-1}S)^{-1}$, where S has orthonormal columns (since $\mathcal{H}_{\mathcal{C}}(\tilde{p}) = [S^T\mathsf{H}_{\mathcal{O}}^{-1}(p)S]^{-1}$).

Define $S^\perp$ so that the matrix $[S|S^\perp]$ is orthogonal. Then the following simple relationship between the Gaussian curvature of the surface and that of its projection exists:

$$\begin{aligned}
&\text{Gaussian curvature of surface} \\
&\qquad = \det(\mathsf{H}) \\
&\qquad = \det\left(S^T\mathsf{H}^{-1}S\right)^{-1} \bullet \det\left(S^{\perp^T}\mathsf{H}S^\perp\right) \\
&\qquad = \begin{pmatrix}\text{Gaussian curvature}\\ \text{of projection}\end{pmatrix} \bullet \begin{pmatrix}\text{Gaussan curvature of } \textit{slice}\\ \textit{perpendicular} \text{ to projection}\end{pmatrix}
\end{aligned}$$

The relation is developed in Appendix 4-D. The 3-dimensional case of the result recently appeared in [73], but while it seems straightforward enough, we have not seen reference to it elsewhere. Indeed, a similar observation was in [73].

The relationship between the two Gaussian curvatures thus involves the extra term $\det(S^{\perp^T}\mathsf{H}S^\perp)$, arising from the Gaussian curvature of a curve obtained as a *slice* of the surface *perpendicular* to the projection defined by S. The problem is how to obtain this extra term given only the Gaussian curvature over the Gaussian sphere and the projection geometry? If we restrict our attention to a point p on the object, all we have is the single number $\mathcal{K}(p)$. This single number in isolation gives no

directional information, and thus no way of distinguishing between projections onto different subspaces. It seems impossible to produce a *local* result (as in (24)) given only knowledge of $\mathcal{K}(p)$ and $\mathcal{P}$. We must somehow take into account surrounding information from the surface, perhaps looking at how the Gaussian curvature changes along the contour generator, or perpendicular to it. By looking at the Gaussian curvature of points perpendicular to the CG perhaps we could 'divide out' the curvature information in that direction. We leave these as questions for future work.

4.5 Conclusions

In summary, in this chapter we have generalized and clarified results on the relationship between local surface curvature of an object and that of its orthogonal projection. In addition to the forward or projection problem, we have given necessary and sufficient conditions for solution of the inverse or curvature reconstruction problem. These conditions, in turn, allowed the formulation of statements directly in terms of the projection subspaces. In Chapter 5 the same underlying algebraic structure shown for local surface curvature projection is shown to exist between an ellipsoid and its projections. The above local considerations were then used to generalize to arbitrary dimensions the representation and projection scheme found in [7, 19]. Our setting and solution, in addition to clarifying existing work, also suggest directions for fruitful future investigation on problems such as reconstruction from finite and misaligned projections.

4-A Notation

n	Dimension of object surface.
m	Dimension of shadow surface.
$\mathcal{O}$	n-dimensional object surface.
$\mathcal{P}$ ($\mathcal{P}_i$)	$(m+1)$-dimensional ((m_i+1)-dimensional) projection subspace.
$\mathsf{H}_{\mathcal{O}}(p)$	The Hessian of $\mathcal{O}$ at p in local coordinates.
$\mathcal{C}$	Boundary of projection of $\mathcal{O}$ in $\mathcal{P}$.
CG	The contour generator. The curve on $\mathcal{O}$ whose image is $\mathcal{C}$.
p	Point on the CG of $\mathcal{O}$.
$\tilde{p}$	Image of p in $\mathcal{C}$.
$\mathsf{H}_{\mathcal{C}}(\tilde{p})$	The Hessian of $\mathcal{C}$ at $\tilde{p}$ in local coordinates.
$\mathcal{T}$	Tangent hyperplane to $\mathcal{O}$ at p.
$\{\hat{t}_1, .., \hat{t}_n\}$	Local coordinate directions at p in $\mathcal{T}$.
$N(p), \hat{y}$	Unit outward normal to $\mathcal{O}$ at p.
$(t_1, .., t_n, y)$	Coordinates of points with respect to the frame $\{\hat{t}_1, .., \hat{t}_n, \hat{y}\}$.
$F(t_1, .., t_n)$	Local representation of $\mathcal{O}$ as $y = F(t_1, .., t_n)$.
$\mathcal{S}$ ($\mathcal{S}_i$)	Subspace of intersection between $\mathcal{T}$ and $\mathcal{P}$ ($\mathcal{P}_i$).
$\{\hat{S}_i\}$	Set of coordinate vectors for $\mathcal{S}$.
$\{\hat{S}_i, N(p)\}$	Set of vectors defining a coordinate frame in $\mathcal{P}$ at p.
$(x_1, .., x_m, y)$	Coordinates of points with respect to the frame $\{\hat{S}_1, .., \hat{S}_m, N(p)\}$.
$f(x_1, .., x_m)$	Local representation of $\mathcal{C}$ as $y = f(x_1, .., x_m)$.
S (S_i)	$n \times m$ (m_i) matrix with columns representations of $\{\hat{S}_i\}$ in frame $\{\hat{t}_1, .., \hat{t}_n\}$.
G	The $n \times n$ target matrix $H_{\mathcal{O}}^{-1}(p)$.
G_i	The i-th observation, $P_i\, G\, P_i$.
P_i	Projector onto projection subspace i, $P_i \equiv S_i S_i^T$.
$\langle \cdot, \cdot \rangle$	The inner product defined on the space S. $\langle A, B \rangle = \mathrm{tr}(A^T B)$.
S	The space of symmetric $n \times n$ matrices with inner product $\langle \cdot, \cdot \rangle$.
$\{M_\ell\}$	Orthonormal symmetric basis for S.
γ	Representation of G in basis $\{M_\ell\}$.
$\mathbf{P}_i$	Image of P_i under the mapping Γ of Chapter 2
g_i	Representation of G_i in basis $\{M_\ell\}$
$\mathbf{g}$	The overall observation vector: $[g_1, .., g_q]^T$.
$\mathbf{P}$	The overall projection matrix: $[\mathbf{P}_1 \vert \mathbf{P}_2 \vert .. \vert \mathbf{P}_q]^T$.
$L(p)$	Transformation to local coordinates $(t_1, .., t_n, y)$ at p.
$\tilde{L}(\tilde{p})$	Transformation to local coordinates $(x_1, .., x_m, y)$ at $\tilde{p}$ in the projection.

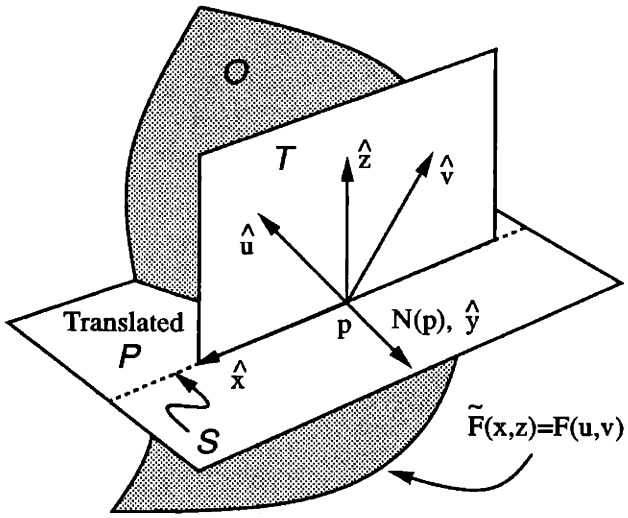

Figure 4-6: Local coordinate system configuration

4-B Proof of Result 24

In this appendix, we shall prove Result 24 relating the curvature of a projection to the curvature of the original surface: $\mathsf{H}_c(\tilde{p}) = (S^T\, \mathsf{H}_{\mathcal{O}}^{-1}(p)\, S)^{-1}$.

The curvature we are discussing here is the Hessian of the surface or curve at a point in some local coordinate system. Recall, that the Hessian is defined as the matrix of second order partial derivatives of the surface, and may be viewed as generating a second order approximation to the surface at the point. In what follows, we shall assume we are working at a point p on the surface of an object and henceforth suppress any mention of that fact to simplify notation (e.g. all partial derivatives are assumed to be evaluated at p). We also assume all surfaces are oriented by their outward normal.

4-B.1 Curvature of the Projection

Assume we are at a point p on the surface that will map to a boundary point of the projection (so p is a point of the contour generator). Also assume that there is a local coordinate system at p oriented in the following way (see Figure 4-6). As in Section 4.2.1, $\hat{y}$ points along the local normal $N(p)$ and the set of m vectors $\{\hat{x}_i\}$ lies in a subspace parallel to the subspace of projection. We complete the set with $n-m$ vectors $\{\hat{z}_j\}$, which lie in a perpendicular subspace. Let y, x, and z, respectively,

be vectors of coordinates along these axes. With this choice, we will be projecting along the z direction onto the x-y plane (the $\{\hat{x}_i\}$ and $\hat{y}$ set of axes together spanning the space of projection). We term these coordinates 'projection coordinates'. In this coordinate system we may represent the surface by the function $y = \tilde{F}(x, z)$. Now the boundary of the projection in these coordinates is defined by the set of equations:

$$y = \tilde{F}(x, z) \tag{4.8}$$

$$\tilde{F}_z = 0 \tag{4.9}$$

where letter subscripts denote vectors or matrices of partial derivatives (for example $\tilde{F}_z = [\tilde{F}_{z_1}, \tilde{F}_{z_2}, \cdots, \tilde{F}_{z_{n-m}}]$ - a row vector). The matrix of second partial derivatives y_{xx} derived subject to the constraints (4.8) and (4.9) is $\mathsf{H}_C(\tilde{p})$, which it is our goal to find.

Start by taking the first partial derivative of (4.8) with respect to x, Applying the chain rule, this gives:

$$y_x = \tilde{F}_x + \tilde{F}_z z_x \tag{4.10}$$

where z_x is the (Jacobian) matrix of partial derivatives. Next, find the unconstrained second partial derivative matrix y_{xx} using (4.10) and the chain rule.

$$y_{xx} = \tilde{F}_{xx} + \tilde{F}_{xz} z_x + \left(\tilde{F}_{zx} + \tilde{F}_{zz} z_x\right) z_x + \tilde{F}_z \left(z_{xx} + z_{xz} z_x\right) \tag{4.11}$$

Now from (4.9), $\tilde{F}_z = 0$, and taking the partial of this equation with respect to x we obtain:

$$\tilde{F}_{zx} + \tilde{F}_{zz} z_x = 0$$

which gives:

$$z_x = -\tilde{F}_{zz}^{-1} \tilde{F}_{zx} \tag{4.12}$$

where $\tilde{F}_{zz}^{-1}$ exists since the surface is strictly convex.

Applying (4.9) and (4.12) to (4.11), we get for the curvature of the projection:

$$y_{xx} = \tilde{F}_{xx} - \tilde{F}_{xz} \tilde{F}_{zz}^{-1} \tilde{F}_{zx} - \left(\tilde{F}_{zx} - \tilde{F}_{zz} \tilde{F}_{zz}^{-1} \tilde{F}_{zx}\right) \tilde{F}_{zz}^{-1} \tilde{F}_{zx}$$

or

$$y_{xx} = \tilde{F}_{xx} - \tilde{F}_{xz}\tilde{F}_{zz}^{-1}\tilde{F}_{zx} \equiv \mathsf{H}_{\mathcal{C}}(\tilde{p}) \tag{4.13}$$

4-B.2 Projection of the Curvature

We shall now relate $\mathsf{H}_{\mathcal{C}}(\tilde{p})$ to $\mathsf{H}_{\mathcal{O}}(p)$. Start by partitioning the set $(t_1, \ldots, t_n)$ of general coordinates in $\mathcal{T}$, defined in Section 4.2.1, so that $\mu = (t_1, \ldots, t_m)$ and $\nu = (t_{m+1}, \ldots, t_n)$. Schematically, we have the relationship between (x, z) and (μ, ν) shown in Figure 4-6. Given these definitions, we may always make the following association between these two sets of coordinates, by proper labeling:

$$\begin{bmatrix} \mu \\ \nu \end{bmatrix} = U \begin{bmatrix} x \\ z \end{bmatrix} \tag{4.14}$$

where $U = [S|S^{\perp}]$ is orthogonal and the columns of S are the representations of the $\{\hat{x}_i\}$ axes in (μ, ν) coordinates. The columns of S thus span the space of the projection in these coordinates. Now given the definition of F in Section 4.2.1, we have the following relationship between F and $\tilde{F}$:

$$F(\mu, \nu) = \tilde{F}(x, z).$$

It is straightforward to show that the Hessians of F and $\tilde{F}$ in the two coordinate systems are related by:

$$\begin{bmatrix} \tilde{F}_{xx} & \tilde{F}_{xz} \\ \tilde{F}_{zx} & \tilde{F}_{zz} \end{bmatrix} = U^T \begin{bmatrix} F_{\mu\mu} & F_{\mu\nu} \\ F_{\nu\mu} & F_{\nu\nu} \end{bmatrix} U \tag{4.15}$$

Now note that (see, for example, [74, pp. 539]):

$$\tilde{F}_{xx} - \tilde{F}_{xz}\tilde{F}_{zz}^{-1}\tilde{F}_{zx} = \left([\, I \mid 0 \,] \begin{bmatrix} \tilde{F}_{xx} & \tilde{F}_{xz} \\ \tilde{F}_{zx} & \tilde{F}_{zz} \end{bmatrix}^{-1} \begin{bmatrix} I \\ \hline 0 \end{bmatrix} \right)^{-1} \tag{4.16}$$

Substituting (4.15) into the right hand side yields:

$$\tilde{F}_{xx} - \tilde{F}_{xz}\tilde{F}_{zz}^{-1}\tilde{F}_{zx} = \left(S^T \begin{bmatrix} F_{\mu\mu} & F_{\mu\nu} \\ F_{\nu\mu} & F_{\nu\nu} \end{bmatrix}^{-1} S\right)^{-1} \tag{4.17}$$

By definition, the Hessian of $\mathcal{O}$ at p in the coordinate system (μ, ν) is:

$$\mathsf{H}_{\mathcal{O}}(p) = \begin{bmatrix} F_{\mu\mu} & F_{\mu\nu} \\ F_{\nu\mu} & F_{\nu\nu} \end{bmatrix}. \tag{4.18}$$

Combining (4.13), (4.17), and (4.18) we obtain:

$$\mathsf{H}_{\mathcal{C}}(\tilde{p}) = \left(S^T \, \mathsf{H}_{\mathcal{O}}^{-1}(p) \, S\right)^{-1} \tag{4.19}$$

We have thus related $\mathsf{H}_{\mathcal{C}}(\tilde{p})$ to $\mathsf{H}_{\mathcal{O}}(p)$ through the projection type operation given in (4.19) and hence the result is proved. ■

4-C Outline of Proof of Corollary 4

The proof of Corollary 4 proceeds as follows. First, we consider just the two hyperplanes defined by P_1 and P_2 and show that $\text{rank}[\mathbf{P}_1^T, \mathbf{P}_2^T]^T = n(n+1)/2 - 1$, just one independent row short of the required number. We then show that a single additional *planar* shadow, defined by P_3, adds this independent observation.

For the first part, note that the number of independent columns of $\mathbf{P} = [\mathbf{P}_1^T, \mathbf{P}_2^T]^T$ equals the number of independent matrices in the set $\{P_i M_j P_i \mid i = 1, 2; j = 1, \ldots, n(n+1)/2\}$, where the $\{M_j\}$ are the elements of an orthonormal basis on S. Thus, showing the first part of the proof may be reduced to a counting argument on the number of independent matrices in the given set, $\{P_i M_j P_i\}$.

We may, without loss of generality, align $n-1$ of the local axes $\{\hat{t}_i\}$ of $\mathcal{T}$ to lie in $\mathcal{S}_1$, say $\{\hat{t}_1, \ldots, \hat{t}_{n-1}\}$. Since $\mathcal{S}_1$ and $\mathcal{S}_2$ will intersect in an $(n-2)$-dimensional subspace in $\mathcal{T}$, we may further align $n-2$ of the above set of axes, say $\{\hat{t}_1, \ldots, \hat{t}_{n-2}\}$, with this intersection space, again without loss of generality. This choice of alignment for the local axes results in the following special form for the projectors $P_1 = S_1 S_1^T$

and $P_2 = S_2 S_2^T$:

$$P_1 = \begin{bmatrix} \mathrm{I}_{n-1} & 0 \\ 0 & 0 \end{bmatrix}, \qquad P_2 = \begin{bmatrix} \mathrm{I}_{n-2} & 0 \\ 0 & X \end{bmatrix}$$

where X is a 2×2, 1-dimensional orthogonal projector with $X_{22} \neq 0$. We couple these observations about the form of P_1 and P_2 with a convenient choice of symmetric basis M_j. Specifically we chose the appropriately normalized set of matrices $\{e_k e_\ell^T + e_\ell e_k^T; k, \ell = 1, \ldots, n\}$ where e_k is the k-th unit vector. The counting argument on the set $\{P_i M_j P_i\}$ is now straightforward, but tedious.

To show the second part, we observe that $P_3 = yy^T$ for some unit n-vector y, with $P_1 y \neq y$ and $P_2 y \neq y$. Thus $P_3 M_j P_3 = \alpha\, yy^T$ for some scalar α and we need only show that yy^T is independent of $\{P_i M_j P_i \mid i = 1, 2; j = 1, \ldots, n(n+1)/2\}$. ■

4-D Relationship between $\mathcal{K}_C$ and $\mathcal{K}_O$

Let $U \equiv [S|S^\perp]$, where $S^T S^\perp = 0$ and U is $n \times n$. Thus U is orthogonal. Using simple determinantal and matrix identities then yields the following:

Gaussian curvature of surface

$$\begin{aligned} &= \det(H) \\ &= \det(U^T \mathsf{H} U) \\ &= \det\left(S^{\perp T} \mathsf{H} S^\perp\right) \bullet \\ &\qquad \det\left(S^T \mathsf{H} S - S^T \mathsf{H} S^\perp \left(S^{\perp T} \mathsf{H} S^\perp\right)^{-1} S^{\perp T} \mathsf{H} S\right) \end{aligned}$$

Again, using a matrix identity, the second term can be rewritten as follows:

$$\begin{aligned} S^T \mathsf{H} S - S^T \mathsf{H} S^\perp \left(S^{\perp T} \mathsf{H} S^\perp\right)^{-1} S^{\perp T} \mathsf{H} S &= \left([\,\mathrm{I} \mid 0\,] \left(U^T \mathsf{H} U\right)^{-1} \begin{bmatrix} \mathrm{I} \\ 0 \end{bmatrix}\right)^{-1} \\ &= \left(S^T \mathsf{H}^{-1} S\right)^{-1} \end{aligned}$$

Thus we obtain the following simple relationship between the Gaussian curvature of the surface and that of its projection:

Gaussian curvature of surface

$$
\begin{aligned}
&= \det(\mathsf{H}) = \det(U^T H U) \\
&= \det\left(S^T \mathsf{H}^{-1} S\right)^{-1} \bullet \det\left(S^{\perp T} \mathsf{H} S^{\perp}\right) \\
&= \begin{pmatrix}\text{Gaussian curvature}\\ \text{of projection}\end{pmatrix} \bullet \begin{pmatrix}\text{Gaussan curvature of } \textit{slice}\\ \textit{perpendicular} \text{ to projection}\end{pmatrix}
\end{aligned}
$$

Chapter 5

Ellipsoids

This chapter focuses on the projection and reconstruction of centered n-dimensional ellipsoids. Ellipsoid reconstruction problems have appeared in the literature, both directly and as bounding approximations. In [9] an ellipse is used as a simple parameterized model for objects in an attempt to recover their eccentricity and orientation from low signal-to-noise ratio tomographic data. In other medical areas ellipsoids are used to model both the shape and volume or area of anatomical parts, such as the heart and spine [75, 76, 77]. In [78, 74] the state of a system is assumed confined to an unknown n-dimensional ellipsoid and the goal is essentially to reconstruct this ellipsoid from observations of its lower dimensional projections. In [79] a group of closely spaced targets in space is observed through a number of passive sensors. The cluster of targets is modeled as an ellipsoid and the observations as projections of it over time. The desire is to find the evolution of the 3-dimensional shape of the ellipsoid.

This problem of ellipsoid projection and reconstruction yields the second physical example involving a symmetric matrix that we examine in detail, the first having appeared in Chapter 4. Chapter 4 showed that a symmetric matrix is related to object shape through local curvature properties. Here a symmetric matrix is used to directly capture global shape information. We expose the common underlying algebraic structure of these two problems in their involvement of a positive semi-definite (PSD) symmetric matrix. The linear projection mapping between symmetric matrices defined in Chapter 2 precisely captures this structure and the results and algorithms of Chapter 3 again yield solutions to the projection and reconstruction

problems.

Outline

In Section 5.1 we associate a centered ellipsoid with an underlying PSD symmetric matrix. This symmetric matrix becomes our representation of the ellipsoid. In Section 5.2 the connection between such an ellipsoid and its orthogonal projection is examined. Our definition of linear projection mapping given in Chapter 2 captures this relationship. Next, the inverse problem of reconstructing a centered ellipsoid from a series of its silhouette projections is treated. A concise characterization of the solution to this problem is given, using the results of Chapter 3. The special case of 1-dimensional ellipsoid projections, resulting in *support* hyperplane observations, is examined in Section 5.3.4. It is shown that for an ellipsoid, observations of support measurements *squared* yield a particularly nice problem. In Section 5.4 the projection and reconstruction of a *dynamic* ellipsoid is examined. Finally in Section 5.5 we show some numerical experiments to illustrate the developments of the chapter.

5.1 Symmetric Matrices

Here we develop the tie between ellipsoids and PSD symmetric matrices. Any symmetric PSD matrix E represents an ellipsoid $\mathcal{E}$ centered on the origin and comprising the set of points given by:

$$\left\{z | z^T u \leq h(u) = \sqrt{u^T \mathsf{E} u},\ \forall u^T u = 1,\ u, z \in R^n\right\} \tag{5.1}$$

Here $h(u) = \sqrt{u^T \mathsf{E} u}$ is the *support function* of the ellipsoid, discussed in Chapter 6. Conversely, for any ellipsoid $\mathcal{E}$ centered at the origin, a unique symmetric PSD matrix can be found such that the description above characterizes the set of points of the ellipsoid. Thus we can represent any ellipsoid $\mathcal{E}$ by its corresponding PSD symmetric matrix E. Note that the ellipsoid is degenerate (i.e. has zero extent) in directions associated with vectors u in the nullspace of E. If E is positive definite, so its inverse exists, then an equivalent definition of the ellipsoid is given by:

$$\left\{z | z^T \mathsf{E}^{-1} z \leq 1,\ z \in R^n\right\} \tag{5.2}$$

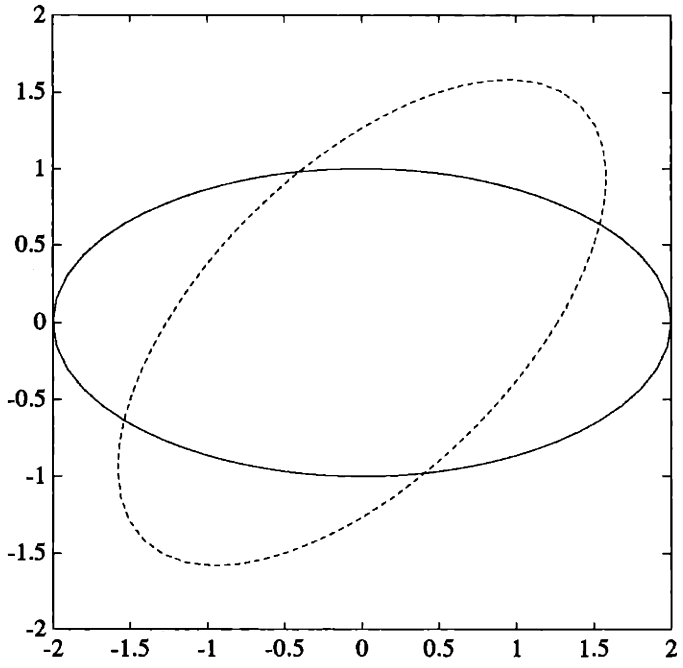

Figure 5-1: Example of ellipse.

The algebraic properties of the symmetric matrix E are reflected in the geometric properties of the corresponding ellipsoid $\mathcal{E}$ in a natural way. If λ_i, ν_i are eigenvalues and eigenvectors of the matrix E, then the principal axes of the ellipsoid are in the directions given by nu_i and the corresponding semi-axes lengths are given by $\sqrt{\lambda_i}$. Thus going back and forth from an ellipsoid to its representation is a simple matter. Further, any linear transformation to the coordinates of the ellipsoid $\mathcal{E}$ as given by $\tilde{z} = Lz$ is reflected by a change of the corresponding matrix E to a matrix E_L given by [74]:

$$\mathsf{E}_L = L\mathsf{E}L^T \tag{5.3}$$

Such transformations include rotation and stretching. For example, consider the ellipse shown by the solid lines in Figure 5-1. The corresponding symmetric matrix is given by:

$$\mathsf{E} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

The ellipse is aligned with the coordinate axes and has semi-axes of length 2 and 1. To obtain an ellipse rotated by ϕ degrees we need only to pick L to be the corresponding

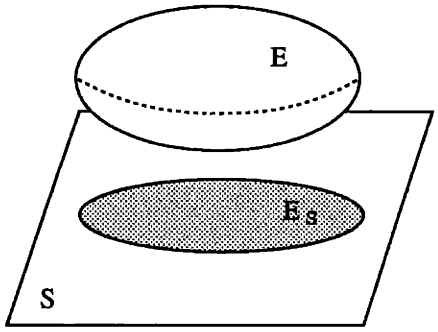

Figure 5-2: Problem definition.

rotation matrix

$$L = \begin{bmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{bmatrix}$$

The matrix E_L corresponding to the rotated ellipse is given by (5.3). For our example ellipse with $\phi = \pi/4$ we obtain:

$$\mathsf{E}_L = \begin{bmatrix} 2.5 & 1.5 \\ 1.5 & 1.5 \end{bmatrix}$$

This rotated ellipse is shown as the dotted lines in Figure 5-1. For convenience in what follows we will often not distinguish between the ellipsoid $\mathcal{E}$ and the matrix E that represents it. In the following we restrict consideration to ellipsoids centered at the origin. There is no loss of generality in this assumption since given a non-centered ellipsoid we may always translate our coordinate system to the ellipsoid origin, thereby recovering the centered case.

5.2 Projections

5.2.1 General Projections

We now consider the orthogonal projection of an n-dimensional ellipsoid $\mathcal{E}$ onto an m-dimensional subspace $\mathcal{S}$ to obtain an m-dimensional shadow object, see Figure 5-2. This projected object will itself be an m-dimensional ellipsoid $\mathcal{E}_\mathcal{S}$ in the subspace $\mathcal{S}$. Let E be the $n \times n$ symmetric matrix representing $\mathcal{E}$ and let E_S denote the corresponding $m \times m$ matrix representing the ellipsoid $\mathcal{E}_\mathcal{S}$ in $\mathcal{S}$. If S is a matrix

whose m columns form an orthonormal basis for the projection subspace $\mathcal{S}$, then we may view the projection operation as a linear transformation of the coordinates of the space. This transformation from the points of the original space z to their coordinates in the projection $\tilde{z}$ is given by the relation $\tilde{z} = S^T z$. Applying (5.3) we find that the relationship between the original ellipsoid, specified by E, and its projection, specified by E_S in the subspace, is simply given by the equation:

$$\mathsf{E}_S = S^T \mathsf{E} S \tag{5.4}$$

Algebraically, this is the same relationship as was found in Result 24 of Chapter 4 between the curvature Hessian of a smooth surface at a point and that of its projection. We examined such relations in Chapter 2. In particular, if we let $X = \mathsf{E}$, $Y = \mathsf{E}_S$, and $A = S$ then the original ellipsoid and its projection are again related through the linear projection mapping between symmetric matrices of Definition 1. This observation forms a tie between our work in Chapters 2, 3, and 4, and again motivates our detailed examination of the projection and reconstruction of symmetric matrices.

Note that we could equivalently represent our projected ellipsoid $\mathcal{E}_S$ by the matrix $\tilde{\mathsf{E}}_S = S\mathsf{E}_S S^T$ rather than E_S. The latter can always be recovered from the former, since S has full column rank. Whereas the $m \times m$ matrix E_S represents a non-degenerate ellipsoid with respect to the subspace $\mathcal{S}$, the singular $n \times n$ matrix $\tilde{\mathsf{E}}_S$ represents a degenerate ellipsoid (one with some axes of zero length) in the original space. The advantage of this form is that the relationship (5.4) between the ellipsoid matrix and its projection then becomes:

$$\tilde{\mathsf{E}}_S = \bar{S} \mathsf{E} \bar{S}$$

where $\bar{S} = SS^T$ is now a true projector onto the subspace $\mathcal{S}$, i.e. is symmetric and satisfies $\bar{S}^2 = \bar{S}$. A consequence is that the projection $\tilde{\mathsf{E}}_S$ is invariant under the particular basis S chosen for the projection subspace $\mathcal{S}$. In what follows we will continue to use the representation E_S because of its more transparent connection to support function information, yet all of our results may be phrased in terms of the projections $\tilde{\mathsf{E}}_S$. In fact, this dual approach is taken in Chapter 4, where support information is not discussed.

5.2.2 1-Dimensional Projections: Support Observations

The special case of 1-dimensional projections is examined here. Such projections are of interest because of their relationship to support function information. The (reduced) support function $h(u)$ of an object is a scalar function of the direction specified by the vector u. It gives a measure of the extent of an object in a particular direction. A detailed discussion may be found in Chapter 6, particularly Definition 4. If the subspace of projection S is 1-dimensional, then S is a unit *vector* and the resulting orthogonal projection of an object is a line segment bounded by the support values $h(S)$ and $h(-S)$. Thus 1-dimensional shadows or projections correspond precisely to a pair of support observations in opposite directions.

The 1-dimensional (scalar) projections E_S of the ellipsoid E have an interesting and useful relationship to the support sample of the original ellipsoid in the direction of the unit vector S. The projected ellipsoid is related to the original one through the equation $\mathsf{E}_S = S^T \mathsf{E} S$. But the right hand side of this equation is just the expression for squared support function $h^2_{\mathsf{E}}(S)$ of the ellipsoid $\mathcal{E}$ in the direction S, as mentioned above [74, pg. 538]. Thus we have that

$$\mathsf{E}_S = h^2_{\mathsf{E}}(S) \tag{5.5}$$

Thus a close relationship exists between the representation E_S of the 1-dimensional projections of an ellipsoid and the support values of the ellipsoid.

We can understand the result (5.5) geometrically. The line segment of the 1-dimensional projection is itself a 1-dimensional ellipsoid, see Figure 5-3. Half the length of the line segment is evidently the semi-axis length. From the algebraic relationship between ellipsoids and the eigenvalues of their representing matrices discussed in Section 5.1, the scalar E_S, representing this projected ellipsoid, must therefore equal the semi-axis length squared. But this semi-axis length is also given by $h_{\mathsf{E}}(S)$, yielding the given relationship.

A consequence of the relationship given in (5.5) is that observations of the support samples of an ellipsoid are equivalent to observations of the quantity E_S for the 1-dimensional projection case. This observation forms another tie between the results of Chapter 6 and the present chapter. In Section 5.3.4 we develop support-based reconstructions of ellipsoids which use these results. Conversely, our linear ellipsoid estimates based on E_S will suggest convenient estimation schemes involving support

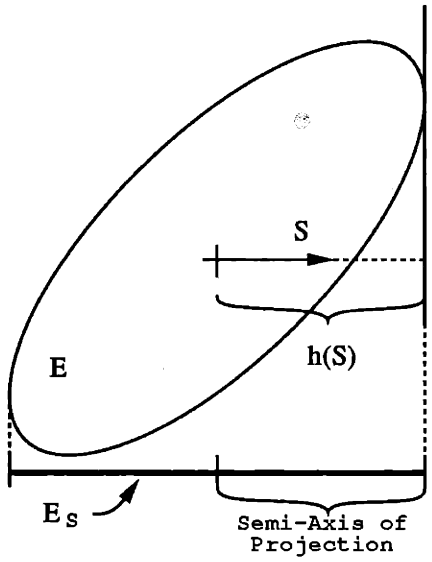

Figure 5-3: Problem definition.

measurements. Naturally, even when the underlying object is not an ellipsoid we may still make use of such methods to find the best fitting ellipsoid, yielding orientation and eccentricity information about an object.

5.2.3 Symmetric Space

Here we use the results of Chapter 2 to represent the relationship between an ellipsoid and its projection given in (5.4) by an equivalent vector equation. Recall from Chapter 2 that the set of $n \times n$ symmetric matrices together with the inner product $\langle A, B \rangle \equiv \mathrm{tr}(A^T B)$ defines an $n(n+1)/2$-dimensional Euclidean space S_n. Recall also that this inner product induces the Frobenius norm on a matrix $\langle A, A \rangle^{1/2} = \|A\|_F$. Following Chapter 2, let $\varepsilon = \Xi(\mathsf{E})$ and $\varepsilon_S = \Xi(\mathsf{E}_S)$ be the vector representations of the matrices E and E_S with respect to the corresponding symmetric bases $\{M_\ell^{(n)}\}$ and $\{M_\ell^{(m)}\}$. The mapping $\Xi(\cdot)$ between a symmetric matrix X and its representation as a vector x in a symmetric basis $\{M_\ell^{(n)}\}$ was defined in Section 2.2. We may thus view the *vectors* ε and ε_S as representing the ellipsoid and its projection, respectively.

By $\tilde{S}$ denote the matrix relating ε and ε_S. From Chapter 2 we know that $\tilde{S}$ is given by the image of S under the map $\Gamma(\cdot)$ of Definition 2. We may now represent

our original relation (5.4) equivalently as:

$$\varepsilon_S = \tilde{S}\varepsilon \tag{5.6}$$

where it is straightforward to show that (see Chapter 2 for more detail):

$$\begin{aligned}(\varepsilon)_j &= \langle \mathsf{E}, M_j^{(n)} \rangle \\ (\varepsilon_S)_i &= \langle \mathsf{E}_S, M_i^{(m)} \rangle \\ (\tilde{S})_{ij} &= \langle M_i^{(m)}, S^T M_j^{(n)} S \rangle\end{aligned}$$

for $1 \leq j \leq n(n+1)/2, 1 \leq i \leq m(m+1)/2$, where $(\cdot)_{ij}$ denotes the ij-th component of the argument. In particular, note that for the case of 1-dimensional projections $m = 1$ so that $\tilde{S}$ is a row vector and, from the arguments of Section 5.2.2, the quantity ε_S is a scalar equal to the support value squared $h^2_{\mathsf{E}}(S)$ in the direction of the vector S. The vector space relationship (5.6) between symmetric matrix representations is the same one seen in (2.6) of Chapter 2 and in (4.3) of Chapter 4.

5.3 Reconstruction

5.3.1 Problem Formulation

In this section we consider the inverse problem of reconstructing an ellipsoid $\mathcal{E}$ from observations of a set of its possibly noisy orthogonal projections onto the subspaces $\mathcal{S}_i$. From the discussion in Section 5.1, we may represent the desired ellipsoid by the symmetric matrix E and its projections onto the subspaces $\mathcal{S}_i$ by the corresponding symmetric matrices $\mathsf{E}_{\mathcal{S}_i}$. For convenience we refer to the projected matrices $\mathsf{E}_{\mathcal{S}_i}$ as simply E_i. If S_i are matrices whose columns form an orthonormal basis for the subspace of projection $\mathcal{S}_i$ (assumed known), then the relationship between the ellipsoid matrix E and its projections E_i is given by (5.4), yielding the following statement of our problem:

Problem 8 (Ellipsoid from Projections) *Determine the $n \times n$ positive semi-definite, symmetric matrix* E, *given q observations of the form*

$$\mathsf{E}_i = S_i^T \mathsf{E} S_i \qquad 1 \leq i \leq q$$

where the matrices S_i have orthonormal columns.

This problem is fundamentally the same as the curvature reconstruction problem, Problem 7 of Chapter 4, and the PSD symmetric matrix reconstruction problem, Problem 3 of Chapter 3, treated earlier. We may thus use the techniques and results developed for those problems here. In particular, we use the vector space relation (5.6) to express each of the observations as

$$\varepsilon_i = \tilde{S}_i \varepsilon$$

where, as for (5.6), ε is the representation of the E, ε_i is the representation of E_i, and $\tilde{S}_i$ is the image of S_i under the map Γ of Chapter 2. Stacking up the observation vectors ε_i into a single vector we obtain the following overall relation:

$$\begin{bmatrix} \varepsilon_1 \\ \hline \varepsilon_2 \\ \hline \vdots \\ \hline \varepsilon_q \end{bmatrix} = \begin{bmatrix} \tilde{S}_1 \\ \hline \tilde{S}_2 \\ \hline \vdots \\ \hline \tilde{S}_q \end{bmatrix} \varepsilon$$

or

$$\mathbf{e} = \mathbf{S}\, \varepsilon \tag{5.7}$$

where $\mathbf{e}$ and $\mathbf{S}$ are defined in the natural way from the stacked observations. Thus, without a semi-definiteness constraint on the reconstruction, Problem 8 is equivalent to: find the unknown vector ε representing the desired ellipsoid, given the observations $\mathbf{e}$ and the geometry specified in $\mathbf{S}$. This formulation is again a standard one in linear estimation. In general the semi-definiteness constraint is needed however. Such issues are discussed in more detail in the next section.

The reader should note that our assumed observations are the projected *ellipsoids* $\mathcal{E}_S$, or equivalently the matrices E_S. Thus, the above formulation has an implicit step of *ellipsoid extraction* from the projections. Any noise in the observations thus man-

ifests itself as noise in the parameters of the projected observed ellipsoid. For many cases this assumption should not pose a significant difficulty. Much work exists, for example, on extracting ellipses from projected planar data [80, 81, 82, 83, 84]. For the case of 1-dimensional projections in particular, fitting the ellipse corresponds to nothing more than extracting the region of support of a line segment, as discussed Section 5.2.2. Finally, we may always view a projection of any dimensionality as a group of (noisy) 1-dimensional projections instead of a single higher dimensional one, thus reducing the problem to the 1-dimensional case. This insight essentially reduces the observation problem to one of boundary point determination. In particular, we could perform the preliminary step of extracting the observed ellipsoids from each projection by using this technique of fitting (lower dimensional) ellipsoids to 1-dimensional support data. We could also directly use all the sets of 1-dimensional observations from all projections simultaneously to directly estimate the desired ellipsoid.

5.3.2 Unconstrained Reconstruction

In this section we consider the solution of Problem 8 without a PSD constraint on the solution. First we consider reconstruction from a consistent set of observations, obviating the need for the PSD constraint. The formulation of (5.7) allows us to easily characterize the unique solutions of Problem 8, as we did in Result 18. Formally we have

Result 26 (Ellipse Solvability) *Problem 8 has a unique solution if and only if the matrix* $\mathbf{S}$ *of (5.7) has rank equal to* $n(n+1)/2$ *(i.e. full column rank). This solution, if it exists, is given by*

$$\varepsilon = \mathbf{S}^L \mathbf{e} \tag{5.8}$$

where $\mathbf{S}^L$ *is any left inverse of* $\mathbf{S}$, $\mathbf{e}$ *is the vector of observations, and* ε *is the representation of the desired matrix* E *with respect to the basis* $\{M_i^{(n)}\}$ *defined in (5.7).*

Noisy Observations

In the case of a set of noisy observations, resulting in a corresponding set of inconsistent equations of the form (5.7), we seek the unconstrained linear least squared error

(LLSE) solution to the set. This estimate $\varepsilon_{\text{LLSE}}$ is obtained as the solution of

$$\varepsilon_{\text{LLSE}} = \arg\min_{\varepsilon} \|\mathbf{e} - \mathbf{S}\varepsilon\|_2^2 \tag{5.9}$$

The corresponding symmetric matrix E_{LLSE} minimizes the quantity:

$$\sum_{i=1}^{q} \|\mathsf{E}_i - S_i^T \mathsf{E}_{\text{LLSE}} S_i\|_F^2$$

This problem is the same as Problem 2 of Chapter 3 and the corresponding solution is obtained by choosing $\mathbf{S}^L = \mathbf{S}^+$, the Moore-Penrose inverse of $\mathbf{S}$, in Result 26. Thus $\varepsilon_{\text{LLSE}} = \mathbf{S}^+\mathbf{e}$ is the desired LLSE estimate without a semi-definiteness constraint on the solution.

To generate an ellipsoid, the matrix E, represented by the vector ε, must be positive semi-definite. The LLSE estimate given by $\varepsilon_{\text{LLSE}} = \mathbf{S}^+\mathbf{e}$ has no such constraint to guarantee this PSD property of the solution. Note, however, that if a PSD matrix is obtained as the LLSE estimate without such a constraint, then clearly it is also the LLSE estimate subject to such a constraint. For many problems the observations are clean enough that the PSD nature of the solution is maintained anyway. A precise statement of this condition was given in Result 20 of Section 3.3.1 in terms of bounds on the allowed perturbations in the observations E_i and the singular values of the matrices $\mathbf{S}$ and E. We interpret these conditions in the context of our ellipsoid problem here.

If the underlying matrix E is a PSD matrix, then Result 20 guarantees that the LLSE estimate will also be PSD if the following condition is satisfied:

$$\sigma_{\min}^{-1}(\mathbf{S}) \sqrt{\sum_{i=1}^{q} \|\delta\mathsf{E}_i\|_F^2} \leq \lambda_{\min}(\mathsf{E})$$

where $\lambda(\cdot)_{\min}$ denotes the minimum eigenvalue value of the argument, $\sigma_{\min}(\cdot)$ the minimum singular value, and the $\delta\mathsf{E}_i$ are differences in the observations from their noiseless values. We would like to have the quantities $\lambda_{\min}(\mathsf{E})$ and $\sigma_{\min}(\mathbf{S})$ large and the quantity $\delta\mathsf{E}_i$ small. The square root term is the easiest to interpret; it corresponds to the overall noise in the observations. Recalling the tie between an ellipsoid $\mathcal{E}$ and the eigenvalues of the corresponding PSD symmetric matrix E, the term $\lambda_{\min}(\mathsf{E})$ can

be seen to be the length of the smallest semi-axis of the ellipsoid. This length might be thought of as a measure of the closeness of the ellipsoid to degeneracy. Finally the term $\sigma_{\min}(\mathbf{S})$ reflects the nearness to singularity of the matrix $\mathbf{S}$. If this quantity is small, the columns of $\mathbf{S}$ are nearly dependent. Since $\mathbf{S}$ captures the observation geometry, such a situation reflects the fact that our observations are nearly linearly dependent, as might happen if we were to use a set of projections on subspaces very close to each other (see e.g. [61] for more detail on the nearness of subspaces). In summary, if the smallest aspect of the underlying ellipsoid is large relative to the noise in the observations, and if our set of observations are well placed, we should be able to use the unconstrained LLSE estimate without the necessity of a PSD constraint on the solution. We demonstrate such unconstrained reconstructions in Section 3.2.

Recursive LLSE

Without the PSD constraint, the solution to (5.9) may easily be implemented recursively. Such a recursive solution is just a recursive least squares formulation, with the state vector of the estimator at time k being the estimate $\hat{\varepsilon}_k$ of the symmetric matrix represented by ε. The relevant equations may be found in Section 3.2.4. Further, if we (somewhat arbitrarily) model the noise in the observations E_S as being Gaussian random variables in each entry, then the linear relationship between the symmetric observation matrix E_S and its representation ε_S ensures that the the noise in the elements of ε_S will also be Gaussian processes (not independent in general). Thus it is a straightforward matter to choose the gains of the recursive estimator to yield the optimal minimum variance filter under such a stochastic interpretation of the noise.

5.3.3 Constrained Reconstruction

While in many instances the unconstrained LLSE solution presented in (5.9) is adequate, there may be situations when, due to incomplete, noisy observations, this estimate is not positive semi-definite. In addition, there may be situations where, because of prior information, we wish to impose constraints on the reconstructed ellipsoid in the form of bounds on its shape. Such bounds might reflect our knowledge of the minimum or maximum breadth of an object we wish to estimate. We investigate such constrained problems in this section.

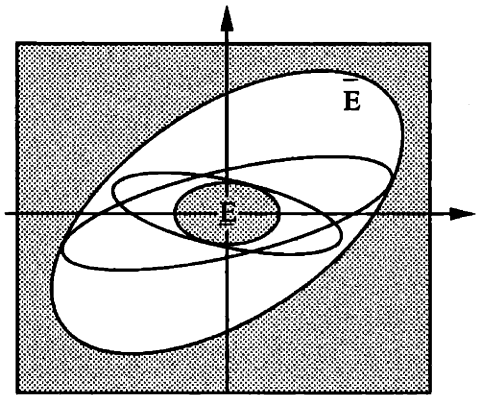

Figure 5-4: Illustration of interval matrix geometry.

Interval Matrix Constraints

Consider the requirement that the reconstructed ellipsoid matrix E lie in the matrix interval given by $\overline{\mathsf{E}} \geq \mathsf{E} \geq \underline{\mathsf{E}}$. By such matrix inequalities we mean that the matrices $(\mathsf{E} - \underline{\mathsf{E}})$ and $(\overline{\mathsf{E}} - \mathsf{E})$ are positive semi-definite. In particular, a PSD constraint is recovered if we choose $\underline{\mathsf{E}} = 0$ and $\overline{\mathsf{E}} = \infty \cdot I$. The algebraic problem of reconstructing a symmetric matrix under such an interval constraint was discussed in Section 3.4, where an algorithm yielding a solution to the problem was given. In the present section we concentrate on the implications of these constraints in the context of our ellipsoid problem.

Such an interval matrix constraint might reflect our knowledge that the underlying ellipsoid lies between the two extreme ellipsoids specified by $\underline{\mathsf{E}}$ and $\overline{\mathsf{E}}$. We know from the discussion of Section 3.4 that in the space of the matrices E, the elements of the interval set are contained in the intersection of two shifted PSD cones, as shown in Figure 3-3. Geometrically, the constraints imposed by the inequality $\overline{\mathsf{E}} \geq \mathsf{E} \geq \underline{\mathsf{E}}$ on the ellipsoid E are natural and easy to interpret. Specifically this condition is satisfied for any ellipsoid contained in the outer extreme ellipsoid $\overline{\mathsf{E}}$ and containing the inner extreme ellipsoid $\underline{\mathsf{E}}$. We illustrate the planar case in Figure 5-4, where the allowed area is the white region and some elements on the boundary of the interval set are shown as ellipses touching the boundary of one extreme or the other.

This geometric interpretation follows from the definition of an ellipsoid given in (5.2) and the positive definiteness of the quantities $(\mathsf{E} - \underline{\mathsf{E}})$ and $(\overline{\mathsf{E}} - \mathsf{E})$. Consider the situation shown in Figure 5-5. If the interval conditions hold, then $\mathsf{E} \geq \underline{\mathsf{E}}$ and $\overline{\mathsf{E}} \geq \mathsf{E}$. But this is true if and only if $\underline{\mathsf{E}}^{-1} \geq \mathsf{E}^{-1}$ and $\mathsf{E}^{-1} \geq \overline{\mathsf{E}}^{-1}$ [38]. Hence, for any given

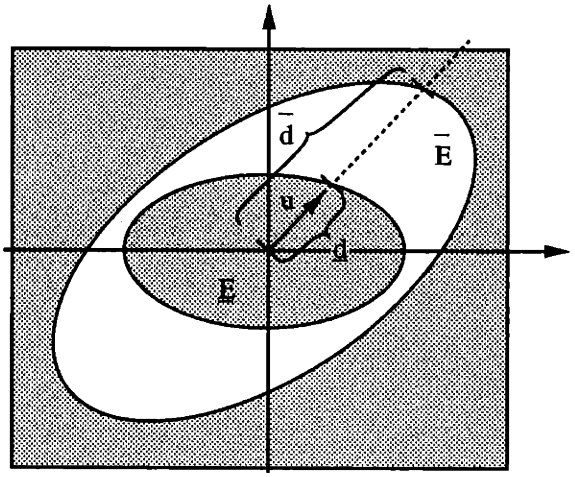

Figure 5-5: Illustration of interval matrix geometry.

unit vector u, an ellipsoid E in the given interval satisfies

$$\overline{d} = \frac{1}{\sqrt{u^T \overline{\mathsf{E}}^{-1} u}} \geq \frac{1}{\sqrt{u^T \mathsf{E}^{-1} u}} = d$$
$$d = \frac{1}{\sqrt{u^T \mathsf{E}^{-1} u}} \geq \frac{1}{\sqrt{u^T \underline{\mathsf{E}}^{-1} u}} = \underline{d}$$

From the definition (5.2) the quantities d, $\underline{d}$, and $\overline{d}$ are just the distances in the direction u to the boundaries of the ellipsoids specified by E, $\underline{\mathsf{E}}$, and $\overline{\mathsf{E}}$, respectively, see Figure 5-5. This argument also holds in the other direction, justifying our geometric interpretation of the interval matrix condition.

Eigenvalue Constraints

Let us consider some special cases of the constraint set $\overline{\mathsf{E}} \geq \mathsf{E} \geq \underline{\mathsf{E}}$. When the extreme matrices are given by a scalar times the identity, so that $\underline{\mathsf{E}} = \underline{\alpha} I$ and $\overline{\mathsf{E}} = \overline{\alpha} I$, the corresponding extreme ellipsoids become nested spheres. This case corresponds to putting simple eigenvalue constraints on the reconstructed matrix E. Such constraints are non-directional since they do not favor one ellipsoid orientation over another. This is reflected in the central symmetry of the extreme ellipsoids. In particular, if $\underline{\alpha} = 0$ and we let $\overline{\alpha} \to \infty$ we again recover the PSD constraint. Since the inner bound is just the origin, note that the (degenerate) ellipsoids corresponding to such PSD matrices may be little more than a line. As expected, these ellipsoids, containing the origin,

exist at the boundary of the set, corresponding to singular matrices E.

Since the structure of the equations of Problem 8 is the same as that treated in Chapter 3, the results and algorithms of Section 3.4 may be applied to obtain the desired constrained LLSE estimates of the symmetric matrix E. In particular, the interval constrained successive hyperplane algorithm will yield an iterative solution to the corresponding matrix interval constrained problem. Such estimates are demonstrated in Section 5.5.

5.3.4 Support-Based Reconstruction

In this section we examine issues arising from a consideration of 1-dimensional projections. Such projections result in support sample observations in complementary directions, as discussed in Section 5.2.2. We show how the consistency constraints of Chapter 6 may be incorporated into our existing constrained reconstruction algorithms to yield improved estimates in certain situations. We also study the special case of planar ellipsoids (ellipses) and support observations. Using the projection parameter space approach of Section 3.2.3 we suggest that working with the squared support function is more natural in some ways than the support function itself.

Before proceeding, let us consider the interpretation of our cost function for this 1-dimensional case. From Section 5.5 we know that for the case of 1-dimensional projections the quantity $S_i^T \mathsf{E} S_i$ corresponds to the squared support function $h_{\mathsf{E}}^2(S_i)$ of E in the direction S_i. Thus our LLSE cost function

$$\sum_{i=1}^{q} \left(\mathsf{E}_i - S_i^T \mathsf{E} S_i \right)^2 = \sum_{i=1}^{q} \left(\mathsf{E}_i - h_{\mathsf{E}}^2(S_i) \right)^2$$

is minimizing the sum of the squared difference between the observed support values squared and our estimated support values squared. Geometrically, this quantity is difficult to interpret in general. For the planar case the formula for the area of a convex object in terms of its support function is given in (6.5) as

$$\text{Area} = \frac{1}{2} \int_0^{2\pi} \left(h^2 - (dh/d\theta)^2 \right) \, d\theta$$

where h is the support function as a function of θ. The term $\int (dh/d\theta)^2 \, d\theta$ on the right hand side of this expression is proportional to the perimeter of the object. Thus

for the planar case our LLSE cost appears to be related to the area deviation between estimate and observation. The extra square appearing in the 2-norm expression for the LLSE would seem to suggest correspondingly much larger weighting on large deviations than a simple area deviation would yield, however. Note that the *maximum* deviation in support value between two objects (the ∞-norm of the difference) corresponds to the Hausdorff metric between the objects, while the corresponding *sum* of the absolute values of the support deviations (the 1-norm) corresponds the perimeter deviation between the objects [85].

Use of Consistent Support

Because of the close ties between our observations and support samples, we can make use of the support consistency results of Chapter 6. In particular, we may use a two-step process of first estimating a consistent set of support measurements and subsequently using these measurements as observations for the reconstruction of an ellipsoid. Such a procedure can yield significantly different and improved estimates over simply imposing a PSD constraint directly on the ellipsoid matrix, particularly when noise levels are high. The reason appears to be that the process of first imposing consistency on the observations functions as a sort of smoothing process, removing some of the extreme variations in the observations. Note that by fitting an ellipsoid to an inconsistent set of support measurements, we also obtain a corresponding set of consistent support values as a consequence, i.e. those of the ellipsoid. Thus the PSD constraint on the ellipsoid matrix E effectively imposes a consistency constraint on the resulting reconstructed set of support values. Here we are imposing this consistency differently.

The first step of estimating a consistent set of support measurements may be simply phrased as a linear inequality constrained least squares problem. If we let the vector $\mathbf{h}_o = [h_1|h_2|\cdots|h_q]^T$ be the vector of noisy inconsistent support observations, then the closest vector of consistent support samples $\mathbf{h}_{\text{consist}}$ is given as

$$\mathbf{h}_{\text{consist}} = \underset{\bar{Q}\mathbf{h}\geq 0}{\arg\min} \|\mathbf{h} - \mathbf{h}_o\|_2^2 \tag{5.10}$$

where the constraint matrix $\bar{Q}$ is defined in (6.11) of Section 6.3.3. This problem is straightforward to solve. Note that choosing the 1-norm in (5.10) makes this a linear programming problem.

We may then use the resulting consistent support values squared as our observation vector $\mathbf{e}$ to be input to a step of ellipsoid fitting. The solution to this step is provided by the algorithms of Section 3.3.3, particularly Algorithm 1. When the noise is large enough to cause inconsistency in the support measurements, this two-step procedure has the advantage of performing a preliminary smoothing on the data. In this way the effect of extreme values are mitigated.

We do not have any optimality results for this procedure, but may make several points. Consider a stochastic interpretation of the problem. If our observations are of the form $h_i^2 + \eta_i$, where η_i are independent identically distributed Gaussian noise samples, then the direct solution of Problem 8 *without* the additional step of consistent support estimation is the maximum likelihood estimate of E. In reality, however, our observations are more likely to be of the form $(h_i + \eta_i)^2$, which are then squared to yield the input to our estimation algorithms. In this case the direct procedure is no longer optimal (though it may still be reasonable). In fact, it is now biased, due to the squaring operation. Thus the imposition of consistency may indeed improve the estimate in this case.

Further, even in the event that the direct solution is optimal (it is always the least square solution), it minimizes a cost criterion based on the *matrix* E. Such optimal reconstructions in terms of E may not always yield geometrically pleasing shapes to the eye. In general, the set of consistent support values will define a nonempty polyhedron. Fitting an ellipsoid to such a nonempty shape usually yields a corresponding nonempty ellipsoid (corresponding to a positive-*definite* matrix E). The singular matrix that may arise from a direct PSD-constrained reconstruction on the raw data yields a degenerate ellipsoid (in the planar case only a line). We illustrate these points in numerical experiments in Section 5.5.1.

Projection Parameter Space

Finally we examine the special case of 1-dimensional projections of planar objects. In particular we use the projection parameter space approach of Section 3.2.3 together with our squared support observations to note certain nice properties of the squared support function.

When our observations are 1-dimensional and the ellipsoid is planar (an ellipse), we may express our projection matrices S as unit vectors parameterized by angle in the plane, $S(\theta) = [\cos(\theta), \sin(\theta)]^T$. For a given ellipse E our squared support

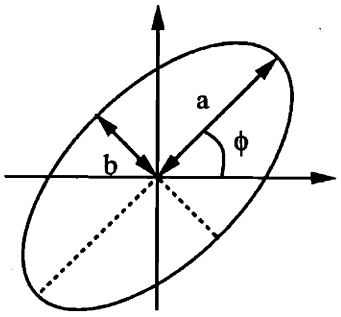

Figure 5-6: Ellipse parameters.

observations may then be expressed as:

$$h^2(\theta) = \beta_1 + \beta_2 \cos(2\theta + \beta_3) \tag{5.11}$$

where $h(\theta)$ is the support value in the direction θ and the β_i are constants depending on the underlying ellipse E. In particular, if $(\mathsf{E})_{ij}$ is the ij-th entry of E then these constants are found using the results of Section 3.2.3 to be:

$$\begin{aligned}
\beta_1 &= \frac{(\mathsf{E})_{11} + (\mathsf{E})_{22}}{2} \\
\beta_2 &= \sqrt{\left(\frac{(\mathsf{E})_{11} - (\mathsf{E})_{22}}{2}\right)^2 + (\mathsf{E})_{12}^2} \\
\beta_3 &= -\arctan\left(\frac{2(\mathsf{E})_{12}}{(\mathsf{E})_{11} - (\mathsf{E})_{22}}\right)
\end{aligned}$$

Suppose the axis lengths of the ellipse are given by a and b and its orientation with respect to the first coordinate axis is given by the angle ϕ, as shown in Figure 5-6. The first term β_1 giving the average of the cosine, is half the trace of the matrix E. Since the eigenvalues of E are the semi-axes lengths squared, this term is given by $\beta_1 = (a^2 + b^2)/2$ and may be thought of a measure of overall size of the ellipse. After some algebra, the second term β_2, expressing the range of the cosine, may similarly be written as $\beta_2 = (a^2 - b^2)/2$ and thus provides a measure of the elongation of the ellipse. The last term, specifying the phase of the cosine, may be shown to be $\beta_3 = -2\phi$, twice the orientation of the ellipse. In this framework the constraint of positive semi-definiteness is reflected in the fact the cosine in (5.11) must always be positive, i.e. that $|\beta_2| \leq \beta_1$.

Since the support function $h(\theta)$ is a periodic function of θ we may represent it as the Fourier series:

$$h(\theta) = \alpha_0 + \sum_{k=1}^{\infty} [\alpha_{ck} \cos(k\theta) + \alpha_{sk} \sin(k\theta)]$$

The first few terms of this series for $h(\theta)$ have natural geometric interpretations. If only the constant term α_0 is nonzero, the corresponding boundary is just a circle centered at the origin. The first order terms define the Steiner point or center of the object, and are zero for a centered object such as our ellipse. Unfortunately, objects containing only the constant and second order terms, while yielding ellipse-like shapes, are not exactly ellipses and have no direct simple interpretation in terms of orientation, eccentricity or other quantities. In addition, centered convex objects having only the constant and second order terms in their support function expansions will have a maximum length-to-width ratio of two [9, pg. 230]. This situation does not appear to change much with the addition of higher order coefficients coefficients either.

Returning to the formula (5.11) for the support function squared, note that it involves only two terms. For a centered object, therefore, (5.11) shows that it is the coefficients of the Fourier series for the support function *squared* that have the desired conic producing property. In particular, the constant and second order terms together precisely define an ellipse, directly giving size, orientation, and eccentricity information. Further, there is no restriction on the eccentricity of the corresponding ellipse, as for a truncated Fourier series of the support function. This observation suggests that it is the support function squared that may be most useful for gross shape extraction and modeling. Specifically, if the center of an object is known or can be estimated (say from the Steiner point), then extracting the constant and second order terms from the Fourier series for the support function squared yields the best fitting ellipse to the boundary data in the sense we have been discussing. This is just another way of solving Problem 8 in the case of 1-dimensional observations. Note that Algorithm 4 of Section 3.3.3 is based on this sine fitting formalism (though not Fourier based) for the planar case and seems to perform well in constrained reconstruction simulations.

The expressions corresponding to (5.11) for the squared support function of a 3-dimensional ellipsoid with projections parameterized by spherical coordinates are

given in Appendix 3-C. For this case, a harmonic surface in the spherical coordinates results. While the structure of these formulas is not as nice as for the planar case, the series is still finite. Thus, as with the planar case, we may find the 2-dimensional Fourier series of the squared support function of an arbitrary centered object and extract only those terms corresponding to the nonzero coefficients for the ellipsoid.

5.4 Dynamic Issues

In this section we treat the problem of generating and estimating a dynamically evolving ellipsoid. These issues are a direct extension of our work on the static case in Section 5.1. We demonstrate a particular symmetric evolution equation and show how we may tailor the characteristics of the ellipsoid evolution through choice of the dynamic matrix. Our previous results allow us to easily represent this evolution in a standard state space form. Because of the symmetry of the matrices, a square root implementation is also possible. Following this examination of the generation problem, we treat the inverse problem of estimating a dynamic ellipsoid from observations of its projections. Of particular interest is the case of 1-dimensional projections, corresponding to observations of the squared support function. Such dynamic ellipsoid problems appear in regard to tracking beating hearts in a series of images, following moving clouds of particles [78], or tracking cells.

5.4.1 Generation

We may animate an ellipsoid by imposing a dynamic relationship on its set of parameters. In particular, evolution of the elements of the matrix E will yield a corresponding dynamically evolving ellipsoid. We need only ensure that the resulting series of matrices $\mathsf{E}(k)$ remain symmetric and positive semi-definite. We thus seek an evolution structure on $\mathsf{E}(k)$ that is simple to implement and yields interesting dynamical behavior, yet maintains symmetry and hopefully positive semi-definiteness.

Dynamic Model

Because of its simplicity, we use the following dynamic model to illustrate the evolution of ellipsoids, with our projection relation (5.4) for observations:

$$\begin{aligned} \mathsf{E}(k+1) &= A_k^T \mathsf{E}(k) A_k + B_k \\ Y_k &= C_k^T \mathsf{E}(k) C_k + W_k \end{aligned} \tag{5.12}$$

where the driving matrix B_k and the observation noise matrix W_k are assumed symmetric to maintain symmetry of the matrix state $\mathsf{E}(k)$. Positive semi-definiteness of $\mathsf{E}(k)$ is assured if $\mathsf{E}(0)$ is positive semi-definite and the matrices B_k are also. To produce independent noise in the entries of Y_k we will make the (independent) entries of W_k independent, zero-mean Gaussian random processes. If B_k is also interpretated as a noise term we may choose its (independent) entries to also be independent zero-mean Gaussian random processes. In this case, if the noise is small, positive semi-definiteness of $\mathsf{E}(k)$ will be likely, but difficult to guarantee. Another possibility is to generate B_k as the square of a random matrix, $B_k = \bar{B}_k^T \bar{B}_k$ with $\bar{B}_k$ random. This approach will assure that B_k remains PSD, but the entries of $\mathsf{E}(k)$ are now not simple functions of a random variable, as they involve products between terms. Similar arguments hold for the matrices W_k as well. In what follows we assume that B_k is a *known* deterministic input matrix. Note that other, more general linear models for the evolution of $\mathsf{E}(k)$ are possible. We use (5.12) for its simplicity, but the algorithms and results of this section hold for more general linear models.

Because the form of both the dynamic and observation equation in (5.12) are identical to (5.4), we may immediately express them as the following equivalent vector equation using the relationship (5.6):

$$\begin{aligned} \varepsilon(k+1) &= \tilde{A}_k \varepsilon(k) + \mathbf{b}_k \\ \mathbf{y}_k &= \tilde{C}_k \varepsilon(k) + \mathbf{w}_k \end{aligned} \tag{5.13}$$

where the vectors $\varepsilon(k)$, $\mathbf{y}_k$, $\mathbf{b}_k$, and $\mathbf{w}_k$ are the representations of the corresponding matrices in (5.12) in a symmetric basis, while $\tilde{A}_k$, and $\tilde{C}_k$ are matrix representations of the operators $A_k^T(\cdot)A_k$ and $C_k^T(\cdot)C_k$ in the symmetric space, obtained under the map Γ given in Definition 2. Thus, instead of the direct equations (5.12), we may equivalently generate the matrices $\mathsf{E}(k)$ using the standard state space equations of

(5.13). Note that the entries of $\mathbf{b}_k$ and $\mathbf{w}_k$ are linear combinations of the respective entries of B_k and W_k.

The form (5.13) is convenient because of the great amount of existing work on such equations. In particular, the observability and controllability of the underlying ellipsoid E follows immediately from the properties of the matrices $\tilde{A}_k$ and $\tilde{C}_k$ together with standard results of linear system theory [86]. An example of the tie between such algebraic properties and the geometric properties of the underlying ellipsoid problem will be seen below when we consider a particular class of dynamic matrices $\tilde{A}_k$.

Square-Root Implementation

Finally, in addition to the above two approaches, we may exploit the symmetry in the problem to obtain a *square root* algorithm for evolution. Such algorithms are used because of their speed and numerical reliability in certain applications. Let $\mathsf{E}(k) = \bar{\mathsf{E}}^T(k)\bar{\mathsf{E}}(k)$ and $B_k = \bar{B}_k^T\bar{B}_k$. Using the singular value decomposition (SVD) we may write:

$$\begin{bmatrix} \bar{\mathsf{E}}(k)A_k \\ \bar{B}_k \end{bmatrix} = U_k \begin{bmatrix} \Lambda_k \\ 0 \end{bmatrix} V_k^T \tag{5.14}$$

where the right hand side is obtained from an SVD of the left hand side. We may then *define* $\bar{\mathsf{E}}(k+1) = \Lambda_k V_k^T$ since

$$\begin{aligned} \mathsf{E}(k+1) &= \bar{\mathsf{E}}^T(k+1)\bar{\mathsf{E}}(k+1) \\ &= \begin{bmatrix} A_k^T\bar{\mathsf{E}}^T & | & \bar{B}_k \end{bmatrix} U_k U_k^T \begin{bmatrix} \bar{\mathsf{E}}(k)A_k \\ \bar{B}_k \end{bmatrix} \\ &= A_k^T\mathsf{E}(k)A_k + B_k \end{aligned}$$

which is the desired relationship.

This form of dynamic update thus requires an SVD of the matrix (5.14). If the dimension n of the underlying space of the ellipsoid is large and the ellipsoid itself is degenerate, corresponding to a large matrix with zero singular values, this procedure could save computations. For our applications however, it seems primarily of academic interest. A useful by-product of these computations is that the axes of the ellipsoid are easily found as the columns of the matrix $V^T\Lambda^{-1}$, where the inverse of Λ is easy to compute since Λ is diagonal.

Shaping Parameters

Here we investigate how certain choices of the dynamic matrices in (5.12) affect the shape of the corresponding ellipsoid. Geometrically we may think of changing the ellipsoid by applying scaling, stretching, and rotation transformations to the underlying coordinate system. For ease of visualization we consider the planar case, though the same arguments hold for arbitrary dimension. We may express such a transformation of coordinates as follows:

$$\tilde{z} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \alpha & 0 \\ 0 & 1/\alpha \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & t \end{bmatrix} z \tag{5.15}$$

The transformation magnifies an object by a factor of t, stretches it in the z_1 direction by α and shrinks it in the z_2 direction by $1/\alpha$, and rotates it by the angle ϕ. This transformation was used in [9] to capture a rich class of object profiles. In particular, it was used in the estimation of object size, eccentricity, and orientation.

The effect on an ellipsoid of applying such a transformation to the coordinate system is given in (5.3). In particular, we may impose this class of transformations dynamically if we choose our matrices A_k as follows:

$$A_k = \begin{bmatrix} t_k & 0 \\ 0 & t_k \end{bmatrix} \begin{bmatrix} \alpha_k & 0 \\ 0 & 1/\alpha_k \end{bmatrix} \begin{bmatrix} \cos(\phi_k) & \sin(\phi_k) \\ -\sin(\phi_k) & \cos(\phi_k) \end{bmatrix} \tag{5.16}$$

Different choices of the parameters of these A_k will result in the application of the corresponding transformation to the ellipse at time point k. The class of dynamic matrices defined by (5.16) is not completely general (stretching along axes other than the coordinate axes is not allowed) but it does capture a large class of transformations. For example, suppose we used an A_k with $\alpha_k = 1$, $t_k = 1$, and $\phi_k = \pi/8$. The resulting ellipse will not change its shape but only rotate by $\pi/8$ radians every step. We show every second step of such a simulation, yielding a tumbling ellipse in Figure 5-7. We may easily extend the class by adding other transformations. For example, by adding another rotation we may impose stretching/compression along arbitrary axes. The generalization to higher dimensions is also straightforward, with the scaling term replaced by a multiple of the identity, the stretching term becoming a diagonal matrix of determinant 1, and the rotation becoming an orthogonal matrix.

An interesting subclass of these transformations is obtained by considering only those that preserve the volume of the ellipse. In cell tracking applications such a

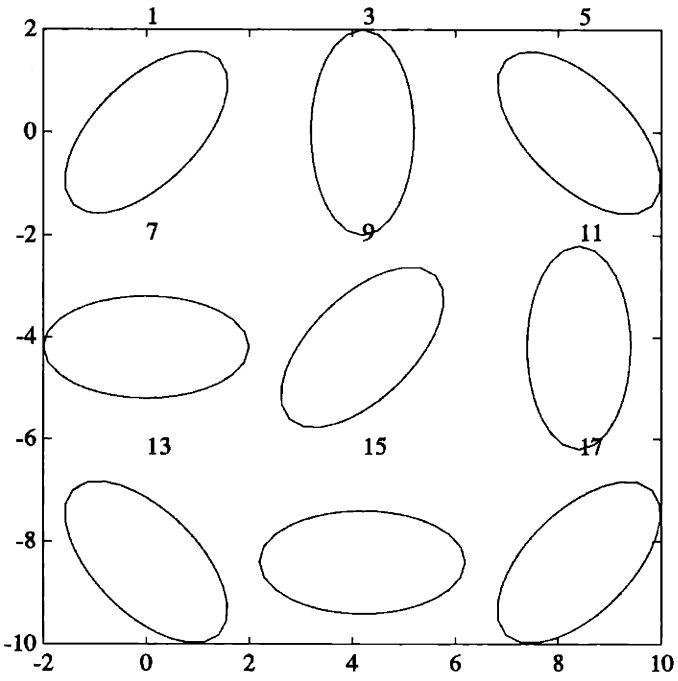

Figure 5-7: Rotating ellipse.

constraint might reflect incompressibility or conservation of mass of a cell undergoing deformational forces in a particular direction. Since the volume of an ellipsoid is a constant times the square root of the determinant of its defining matrix E [46], any transformation preserving the determinant of E achieves the desired goal. For the class of dynamic matrices under consideration, this restriction corresponds to requiring that the scaling term t be set to 1. An example of such a case with $\alpha = 3/4$, $\phi = \pi/8$, and $t = 1$ is given in Figure 5-8, where we have shown every second step of the sequence. This choice corresponds to compressing along the first coordinate axis by a factor of 3/4 and stretching along the second coordinate axes by 4/3 in addition to rotation by $\pi/8$ at each step (The envelope of ellipses traced out in this fashion itself appears to be an ellipse!).

5.4.2 Estimation

In this section we consider the problem of estimating the state of a dynamically evolving ellipsoid. First we recast this problem in terms of the formulation given in (5.7), obtaining a batch method of solution. The inclusion of constraints for this

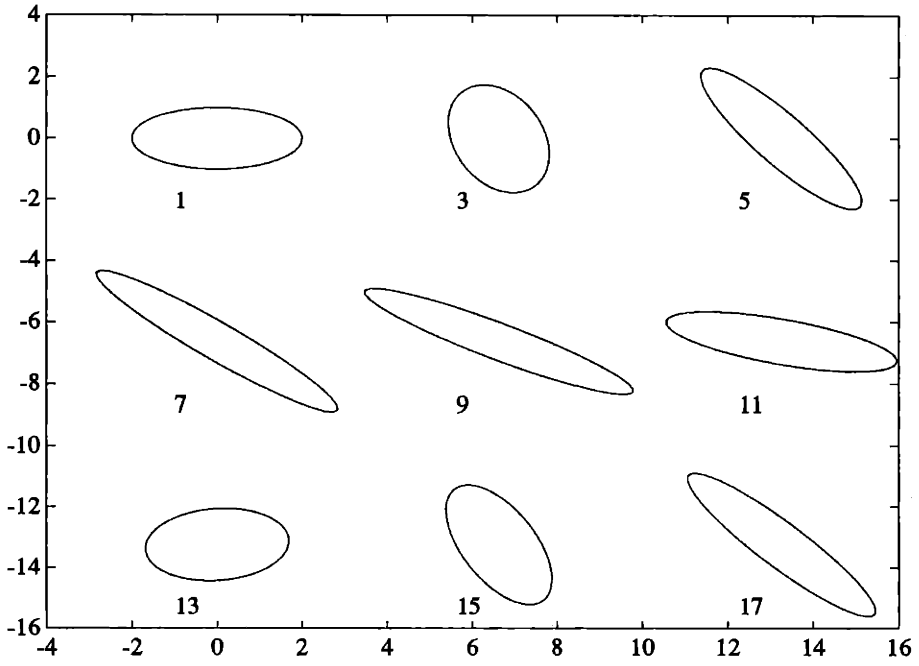

Figure 5-8: Deforming ellipse.

case is straightforward from our work so far. We briefly point out the possibility of recursive solutions to this problem.

Batch Methods

Certainly one way of solving the problem of estimating the state of the ellipsoid in (5.12) is to stack up our observations using the form of the equations given in (5.7), lumping the dynamics and input into the output matrices and observations, respectively. Doing this operation yields a batch formulation of the problem. In particular, we obtain the following equivalent linear equation for the initial state:

$$\bar{\mathbf{y}} = \bar{\mathbf{A}}\varepsilon(0) + \bar{\mathbf{w}} \tag{5.17}$$

where the matrices $\bar{\mathbf{y}}$ and $\bar{\mathbf{A}}$ are given by:

$$\bar{\mathbf{A}} = \begin{bmatrix} \tilde{C}_0 \\ \tilde{C}_1\tilde{A}_0 \\ \tilde{C}_2\tilde{A}_1\tilde{A}_0 \\ \vdots \\ \tilde{C}_r\tilde{A}_{r-1}\cdots\tilde{A}_0 \end{bmatrix} \tag{5.18}$$

$$\bar{\mathbf{y}} = \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_r \end{bmatrix} - \begin{bmatrix} 0 \\ \tilde{C}_1\mathbf{b}_0 \\ \vdots \\ \tilde{C}_r\left(\mathbf{b}_{r-1} + \tilde{A}_{r-1}\mathbf{b}_{r-2} + \cdots + (\tilde{A}_{r-1}\cdots\tilde{A}_1)\mathbf{b}_0\right) \end{bmatrix}$$

and the observation noise vector $\bar{\mathbf{w}}$ is given by:

$$\bar{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_0 \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_r \end{bmatrix}$$

To find a unique initial state $\mathsf{E}(0)$, and thus solution to (5.12), requires that the matrix $\bar{\mathbf{A}}$ given in (5.18) have full column rank. In this case, the unconstrained LLSE estimate of the initial ellipsoid state is given by $\hat{\varepsilon}(0)_{\text{LLSE}} = \bar{\mathbf{A}}^+\bar{\mathbf{y}}$, where $\bar{\mathbf{A}}^+$ is again the Moore-Penrose inverse of $\bar{\mathbf{A}}$. The corresponding unconstrained LLSE estimate at any other time is obtained by using this estimate of the initial state as an initial condition to the equation (5.12) or (5.13).

Since $\varepsilon(0)$ represents a symmetric matrix, to obtain constrained estimates of the initial state we need only combine the constrained reconstruction methods of Section 5.3.3 or 3.3 with the normal equations associated with (5.17). Constrained reconstructions at other times may be found by simply shifting the time origin in (5.17). The solutions of such problems are straightforward given our previous development.

Observability

The existence of a unique solution to (5.17) (or a time shifted version of it if our interest is at other than $k = 0$) required full rank of the matrix $\bar{\mathbf{A}}$ in (5.18). This matrix will be recognized as the observability matrix of the linear dynamical system given by (5.12). Thus our rank requirement for solution of (5.17) is really nothing more than a statement of observability of the corresponding dynamical system. Such

observability is straightforward to check for a given problem using (5.18) and often reflects geometric properties of the problem.

For example, suppose our lower dimensional views are fixed so that $\tilde{C}_k = \tilde{C}$ (non-square) and that we choose the dynamic matrices A_k in (5.16) such that $\phi = 0$ and $\alpha = 0$, corresponding to uniform shrinking of the ellipsoid with no rotation or stretching. The corresponding matrix of (5.13) then becomes $\tilde{A}_k = tI$, where I is the identity. The above observability matrix for this case is given by

$$\begin{bmatrix} \tilde{C} \\ t\tilde{C} \\ t^2\tilde{C} \\ \vdots \\ t^{r-1}\tilde{C} \end{bmatrix}$$

which is clearly rank deficient. Thus we cannot reconstruct an ellipsoid which is uniformly shrinking from a single fixed lower dimensional viewpoint. The problem geometrically is that we get no information about the ellipsoid perpendicular to the projection subspace. The presence of a rotation or stretching term would yield such information as would changing our view by allowing $\tilde{C} = \tilde{C}_k$.

Recursive Approaches

A batch method of estimating a dynamic ellipsoid was provided by the formulation of (5.17). We may also implement this least squares solution recursively. The equations for this dynamic case are given by:

Prediction Step

$$\begin{aligned} \hat{\varepsilon}(k+1|k) &= \tilde{A}_k\hat{\varepsilon}(k|k) + B_k \\ P_{k+1|k} &= \tilde{A}_k P_{k|k}\tilde{A}_k^T \end{aligned}$$

Update Step

$$\hat{\varepsilon}(k|k) = \hat{\varepsilon}(k|k-1) + K_k\left[\mathbf{y}_k - \tilde{C}_k\hat{\varepsilon}(k|k-1)\right]$$

$$\begin{aligned} P_{k|k} &= \left[I - K_k \tilde{C}_k\right] P_{k|k-1} \\ K_k &= P_{k|k-1} \tilde{C}_k^T \left[\tilde{C}_k P_{k|k-1} \tilde{C}_k^T + R_k\right]^{-1} \end{aligned}$$

where R_k is the weight placed on observation k. In particular, $\hat{\varepsilon}(k|k)$ in the above recursive formulas minimizes the quantity:

$$\hat{x}_0^T P_0 \hat{x}_0 + \sum_{i=1}^{q} \left[\mathbf{y}_i - \tilde{C}_i \hat{\varepsilon}(k|k)\right]^T R_i \left[\mathbf{y}_i - \tilde{C}_i \hat{\varepsilon}(k|k)\right]$$

where $P_0 = P_{0|-1}$ is the uncertainty in the initial estimate. For the unweighted LLSE case $R_k = I$. To downweight the initial guess, P_0 should be chosen large. Stochastic interpretations result if the weighting matrices R_k are chosen to be the observation error covariance matrices [60]. For example, under the assumption of independent Gaussian noise for the entries of W_k and hence $\mathbf{w}_k$, these formulas yield the Kalman filter for the system (5.13), resulting in the the maximum likelihood estimate of the initial ellipsoid $\mathsf{E}(0)$.

The above recursion yields a solution to the unconstrained problem. We know of no optimal recursive solution to the constrained problem of estimating a PSD initial matrix for the system in (5.12). Recall that if the unconstrained estimate is positive semi-definite, then it must also be the optimal PSD constrained estimate. Of course, we may take the ad hoc approach of projecting the unconstrained LLSE estimate onto the constraint set when such an estimate is desired. Such a procedure is suboptimal but may yield reasonable results for some cases.

5.5 Experiments

In this section we present the results of several simulations to demonstrate the important features of the different reconstruction schemes. We limit ourselves to planar examples and 1-dimensional support observations here for ease of visualization, though the demonstrated procedures and methods demonstrated are valid in arbitrary dimensions.

First we examine the case of reconstructing a static ellipse from a series of its noisy projections. We compare unconstrained to matrix-interval-constrained reconstructions, showing how the addition of constraints, reflecting prior knowledge, may

aid a reconstruction. In particular, we examine how such algebraic constraints may manifest themselves geometrically. The special case of eigenvalue constraints, including PSD constraints, is also examined. The results of Chapter 6 are then applied to impose consistency on our support-based measurements. We demonstrate the two-step process described in Section 5.3.4 of first estimating a consistent set of support measurements and subsequently using these measurements as observations for the reconstruction of an ellipse. Such a procedure may yield significantly different and improved estimates, compared to simply imposing a PSD constraint directly on the ellipse matrix, particularly when noise levels are high.

Next we examine the reconstruction of dynamic ellipses. We restrict consideration to the class of dynamic matrices defined in (5.16). Typical elements from this class are demonstrated, showing their effect on size and shape change parameters. We then show reconstructions demonstrating how knowledge of the evolution of an ellipse allows us to reconstruct it from knowledge of only a single spatial projection over time.

Assumptions

In the numerical experiments of this section it is assumed that the center of the ellipse is known, as throughout this chapter. For convenience all examples use 2-dimensional ellipsoids (ellipses) and 1-dimensional projections. As discussed in Section 5.2.2, such ellipse projections are synonymous with support measurements, our projections being the *squared* support measurements. Thus we are effectively demonstrating the reconstruction of ellipses from noisy support data.

Our constrained and unconstrained estimates developed in Section 5.3 for 1-dimensional support measurements are based on observations of the noisy *squared* support data, $h^2 + \eta$, with η as some sort of noise. In spite of this, our illustrations of these methods throughout this section will be based on models generated using the more physically realistic situation of directly observing the noisy support values, $h + \eta$, where η is a zero mean Gaussian noise process of intensity σ (notation $\eta \sim N(0, \sigma^2)$). The input to our estimation schemes then becomes $(h+\eta)^2$. Our LLSE reconstructions will, of course, still yield estimates $\hat{h}$ which minimize $\sum[\hat{h}^2 - (h+\eta)^2]$ but these estimates will now be biased and hence suboptimal. In particular, these reconstructions will no longer correspond to the maximum likelihood estimates of h. Good reconstructions are still provided by these methods, however, illustrating their robustness to such violations of the underlying model assumptions.

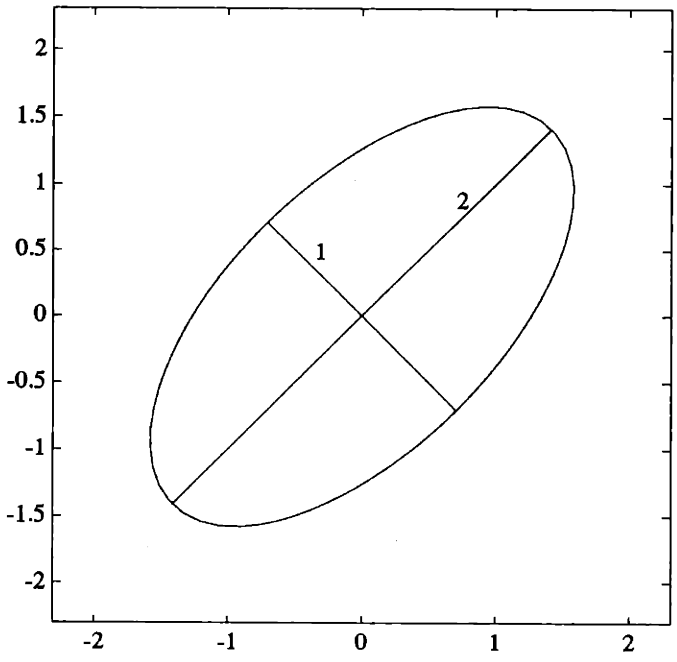

Figure 5-9: Underlying ellipse.

5.5.1 Static Ellipse Reconstruction

The underlying ellipse used throughout this section is shown in Figure 5-9. Its semi-axes are of lengths 1 and 2, with the major axis inclined at $\pi/4$ radians to the first coordinate axis. The corresponding matrix E is given by:

$$E = \frac{1}{2}\begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix} \tag{5.19}$$

Small-Noise Case

First we take 10 equally spaced support samples of the ellipse in Figure 5-9 and add $N(0, .25)$ independent Gaussian noise samples to these points. The resulting observations are shown together with the underlying ellipse in Figure 5-10. Since the axis lengths are 2 and 1, the noise level is relatively small in this example.

In Figure 5-11 three different reconstructions are shown. In the upper left of the figure the unconstrained LLSE solution obtain by using (5.8) is shown. The upper right shows the corresponding PSD constrained reconstruction. This reconstruction is identical to that obtained without the PSD constraint, showing that for this small

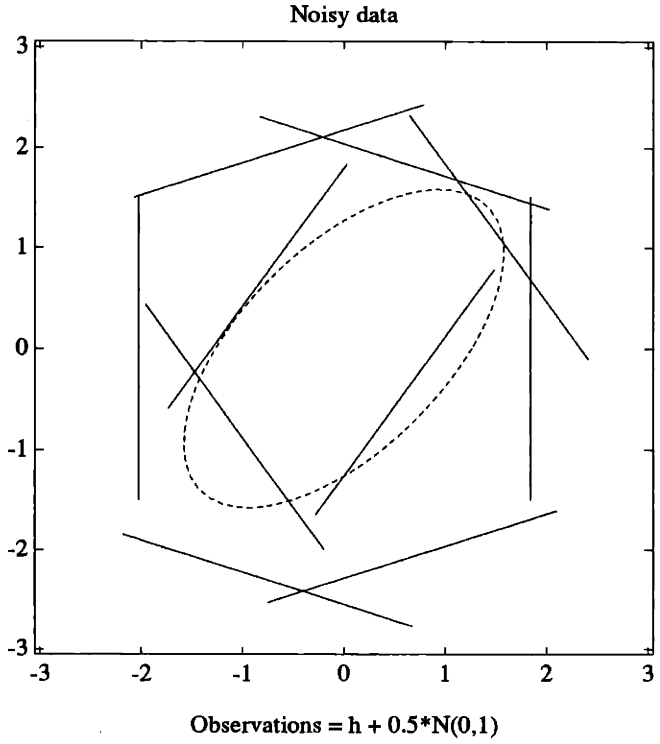

Figure 5-10: Noisy support data ($N(0, .25)$ level noise).

noise case the PSD constraints were not active. The bottom right plot shows inner and outer bounds of an interval-constrained reconstruction. The constraints are arbitrary and only meant to show how such constraints may be included. The corresponding reconstruction is shown in the lower left hand corner. This estimate appears to be the best, demonstrating the effectiveness of correct prior information. If we had no directional information as to the orientation of the ellipse we could replace the ellipse bounds by circles, with no such inherent directional bias. Such circular constraints correspond to *eigenvalue* constraints on the reconstructed matrix.

High Noise Case

Now we repeat the experiment, but with a higher noise level in our observations, this time adding $N(0, 4)$ independent Gaussian noise samples to the support observations. In Figure 5-12 the resulting noisy support data are shown. The noise level is quite high this time, being on the order of axis lengths. In particular there appear to be one or two outliers in the group.

In Figure 5-13 the same three reconstructions provided before are shown. In the upper left of the figure the unconstrained LLSE solution is displayed. The estimated

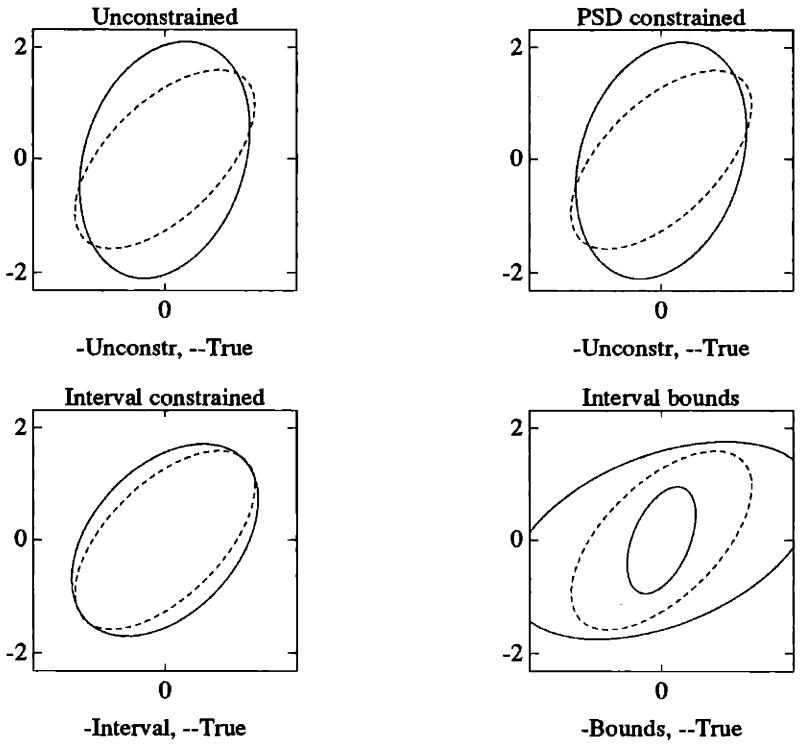

Figure 5-11: Reconstructions.

matrix for this example is not positive semi-definite, having eigenvalues at -1.8 and 12.2. The corresponding figure really does not make sense to draw and certainly does not resemble an ellipse. The curve that is displayed is actually a hyperbola (a different conic section), obtained because we used the ellipse definition given in (5.2) to produce these plots. The upper right plot shows the corresponding PSD-constrained reconstruction. This time the PSD reconstruction yields the degenerate ellipse given by a line, corresponding to the fact that the "negative axis" corresponding to the negative eigenvalue was set to zero. While yielding a PSD matrix, the effect is not particularly appealing geometrically. Again, the bottom right plot shows inner and outer bounds of the same interval constrained reconstruction. The corresponding reconstruction is shown in the lower left hand corner. Again, this estimate appears to be the best.

Consistent Support

Here we demonstrate the reconstruction of consistent measurements, as described in Chapter 6 as well as their use in the two step reconstruction process described in Section 5.3.4. First, in Figure 5-14 we show the closest consistent support vector to

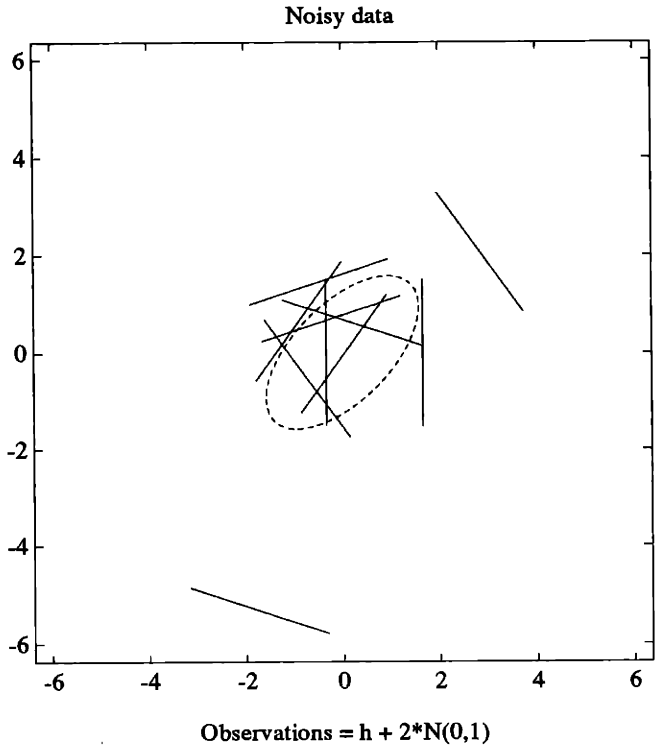

Figure 5-12: Noisy support data ($N(0,4)$ level noise).

the noisy support data shown in Figure 5-10. This consistent support vector is the solution of the problem (5.10) of Section 5.3.3. Such issues of support consistency are discussed in more detail in Chapter 6. Finding this consistent support data corresponds to the first step of the 2 step scheme suggested in Section 5.3.4. Note that the consistent reconstruction of the support data has several points where multiple support lines intersect, showing that this support vector is on the boundary of the consistent set. This interpretation follows from the intersection arguments discussed Section 6.3.

The above consistent support data is now used to generate an ellipse estimate. This unconstrained LLSE estimate based on the consistent data is shown in Figure 5-15. Since the noise is low for this case, there does not appear to be much difference between this estimate and the earlier unconstrained estimate based on the noisy data and shown in the upper left of Figure 5-11.

Next we examine the high noise case shown in Figure 5-12. The equivalent consistent support vector to for this data is shown in Figure 5-16. Compared to the raw inconsistent data in Figure 5-12 the effect of imposing consistency is quite noticeable. In particular the extreme points have been greatly mitigated.

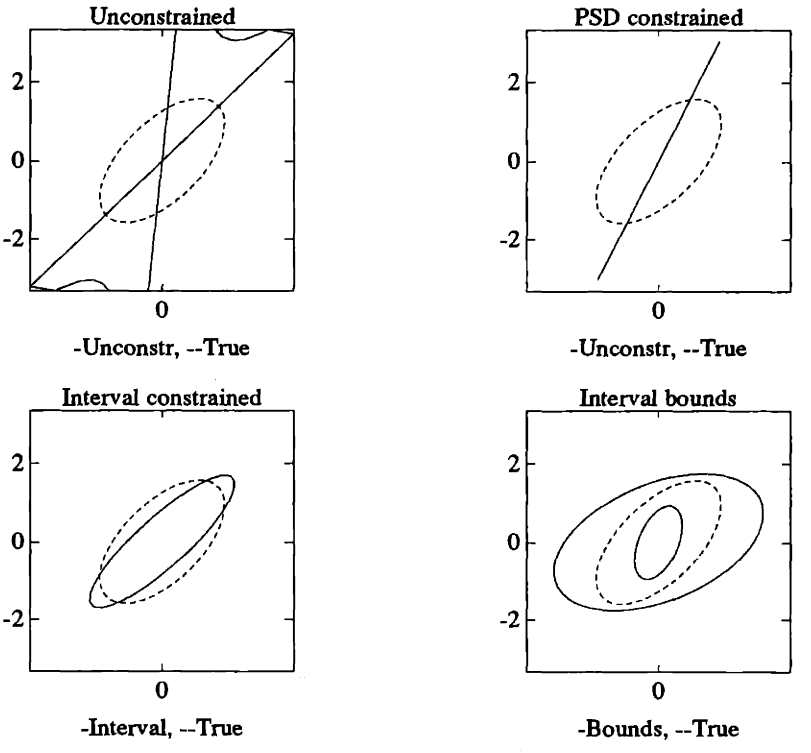

Figure 5-13: Reconstructions.

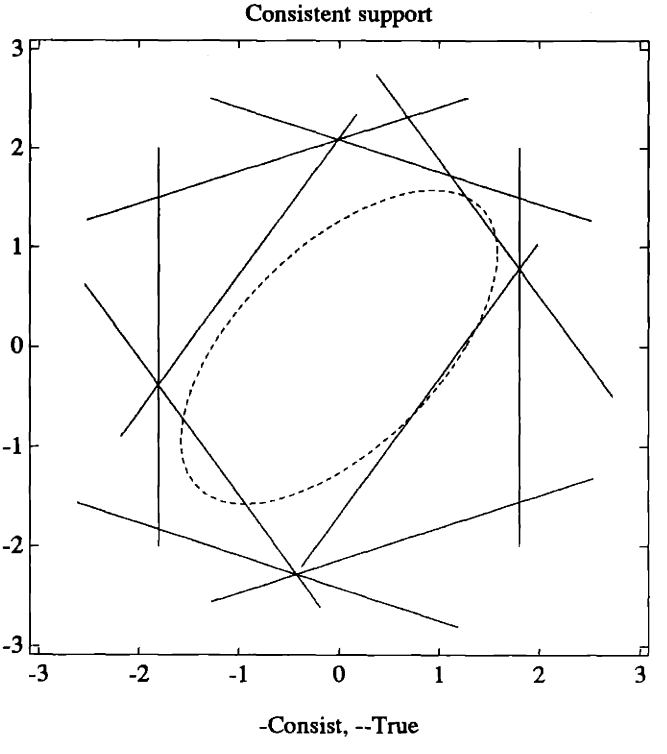

Figure 5-14: Consistent support vector for data in Figure 5-10.

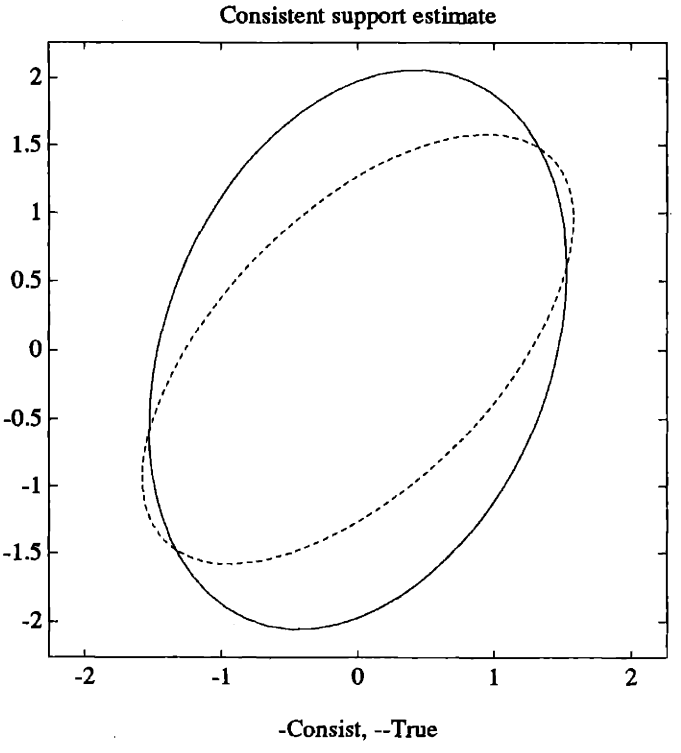

Figure 5-15: Unconstrained LLSE estimate based on consistent data of Figure 5-14.

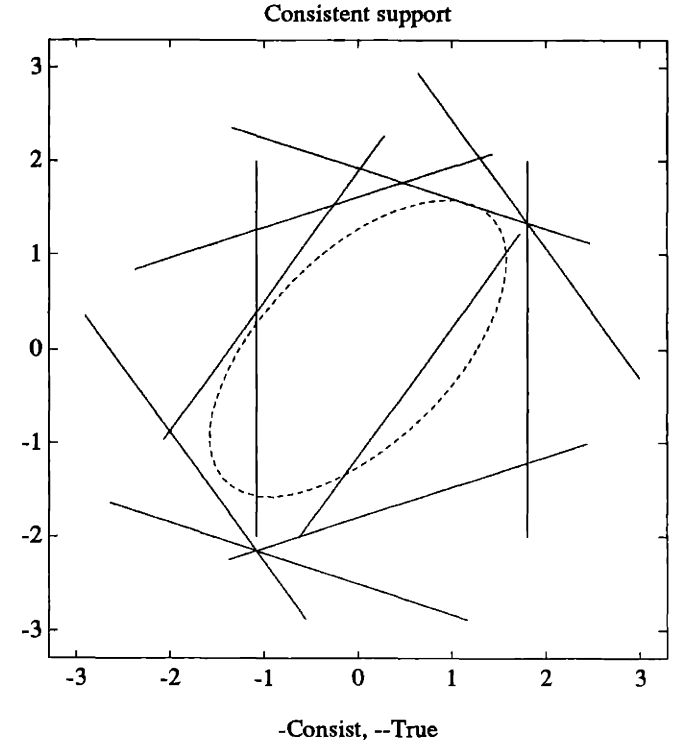

Figure 5-16: Consistent support vector for data in Figure 5-12.

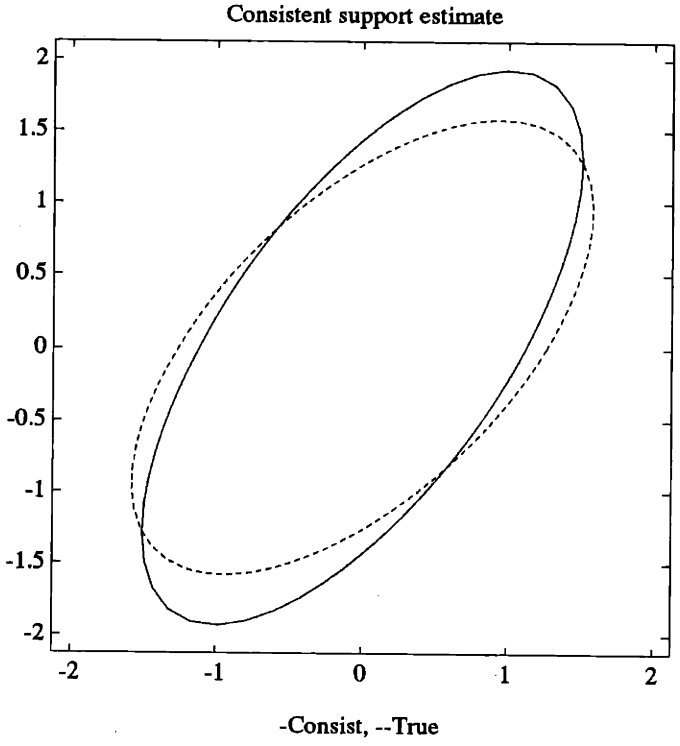

Figure 5-17: Unconstrained LLSE estimate based on consistent data of Figure 5-16.

Using this consistent data in an unconstrained LLSE ellipse reconstruction yields the result shown in Figure 5-17. This consistent estimate is much better than the corresponding unconstrained ellipse estimate using the raw data (shown in the upper left of Figure 5-13), or even the PSD constrained estimate (shown in the upper right of Figure 5-13). Thus, for this high-noise case, the potential value of using consistency of the support data in addition to direct constraints on the ellipsoid itself is demonstrated.

5.5.2 Dynamic Ellipse Reconstruction

In this section we consider *dynamic* ellipse reconstruction form noisy support data. The generation of such an ellipse was discussed in Section 5.4.1, where a particular class of dynamic matrices was defined and demonstrated. For this example we have chosen a constant dynamic matrix $A_k = A$ with the parameters defined in Section 5.4.1 chosen as $t = .8$, $\alpha = .9$, and $\phi = \pi/15$. We use a periodic driving term B_k in (5.12). This drive is shown in Figure 5-18 through half of its cycle. The corresponding ellipse state for this choice of drive and dynamic matrix is shown for every

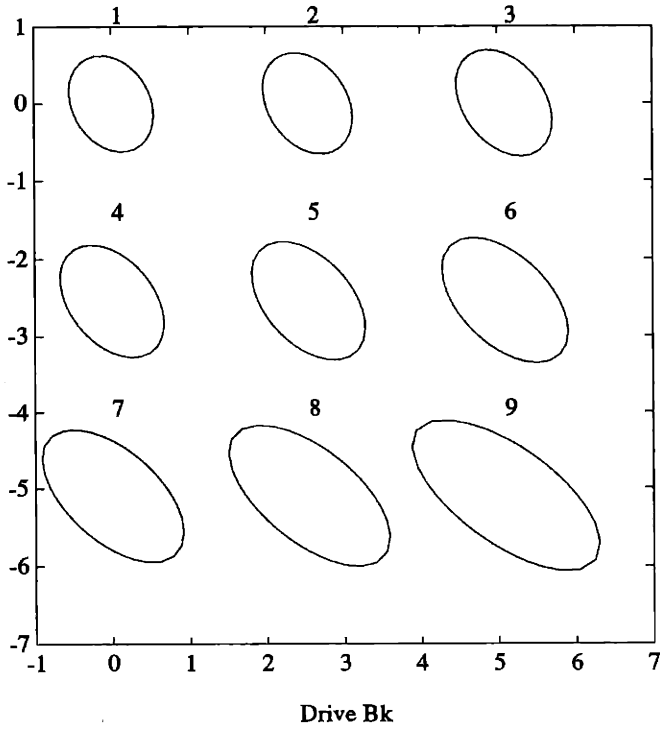

Figure 5-18: Driving term for the dynamic ellipse.

third time point in Figure 5-19. One can imagine similar constructions to model a beating heart, for example.

At each time point the ellipse is projected onto the first coordinate axis, giving a corresponding output matrix in equation (5.12) of $C_k = [1 \quad 0]^T$. From these projections, the support is obtained and $N(0,1)$ noise is added to it, as for the static cases discussed above. Then, using this noisy support data, the original ellipse is reconstructed using recursive least squares with an initial condition given by the matrix E in (5.19). This reconstruction is shown in Figure 5-20. As can be seen from the figure, despite measurements which are quite noisy the estimate tracks the dynamic ellipse after about 7 time steps.

Note that for 1-dimensional fixed projections of a dynamic object, consistency at a given point in time is imposed by default (since there is only a single projection). Thus the 2-step procedure involving consistent support reconstruction illustrated in Figures 5-15 and 5-17 of the previous section appears to be inapplicable in the current setting. Over the entire time sequence, however, we might consider combining the evolution dynamics with the consistent support constraints of Chapter 6 to generalize the 2 step procedure we proposed for the static case. For example, if the dynamics

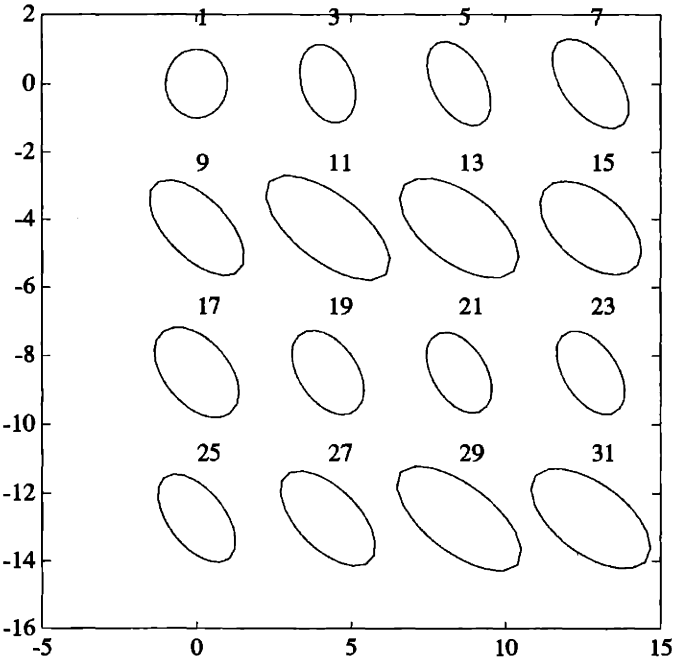

Figure 5-19: Ellipse state.

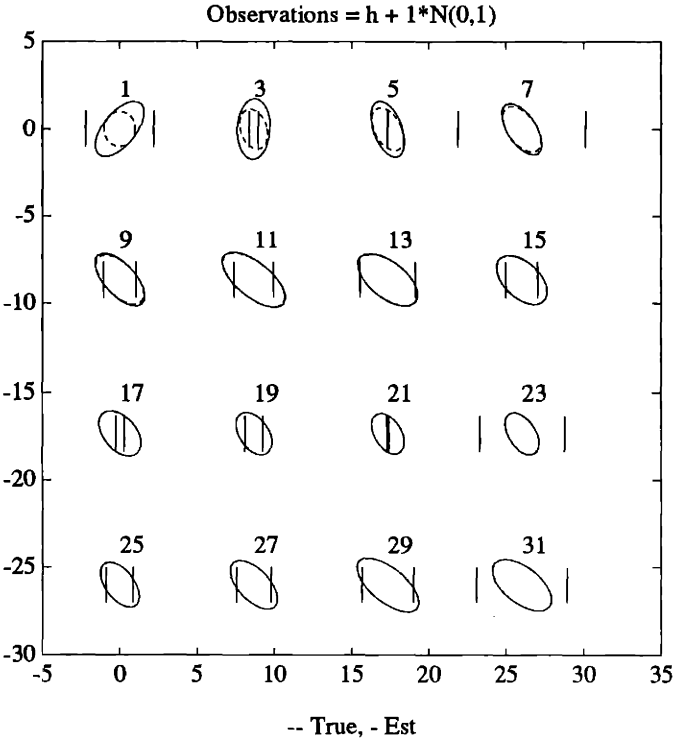

Figure 5-20: Dynamic reconstruction of ellipse of Figure5-19.

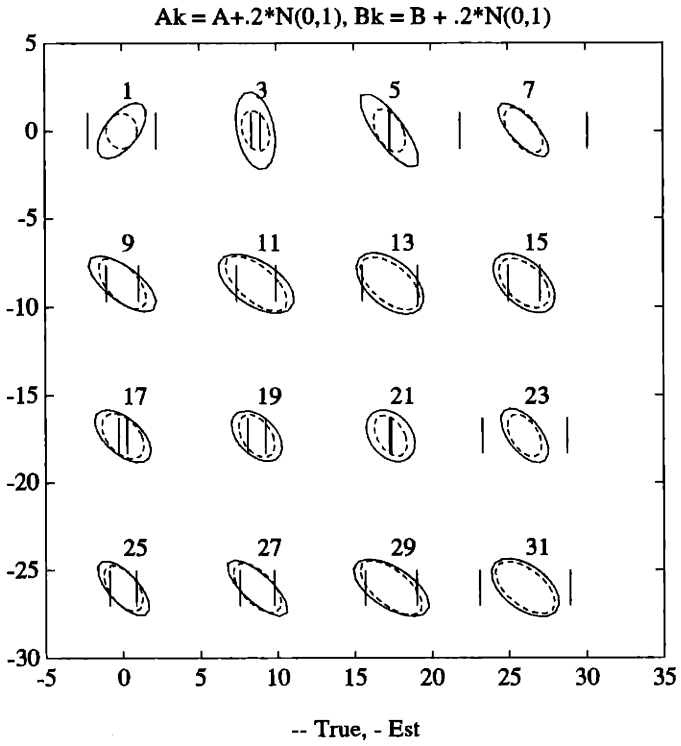

Figure 5-21: Reconstruction of dynamic ellipse with imperfect knowledge of dynamics and drive

imposed by A_k are just a rotation, then there is an equivalent formulation of the dynamic problem as a static one involving multiple views of a fixed ellipsoid (the initial ellipsoid state). This situation is precisely the case used for the 2 step procedure treated earlier. We may thus impose consistency on the equivalent observations and then use these consistent observations to reconstruct the ellipsoid. An interesting question is how to interpert such a procedure in terms of the original problem and how we may implement it recursively. Since the consistency problem is a linear inequality constrained least squares problem (LSI), such issues seem related to the development of a recursive LSI procedure. In the case of more general dynamics, it seems that the corresponding projections of the static problem will not be simple orthogonal projections anymore.

In the above example we assume that we have perfect knowledge of the dynamics and drive and the reconstruction seemed quite good. For interest, in Figure 5-21 we show the effect of adding $N(0, .04)$ noise to our assumed dynamic matrix and drive terms during the reconstruction. This case shows what the effect of imperfect knowledge of the dynamics and drive is on the reconstruction. This noise level is on

the order of these terms, yet the reconstruction is not unduly effected. Apparently the reconstruction is quite robust to such deviations from assumptions. We leave further analysis of these problems for future work.

Chapter 6

Support Measurements: Consistency and Curvature

6.1 Overview

In this chapter we shall investigate some problems and issues arising in the use of support sample observations for the reconstruction of geometric objects. Support samples result from measurements of the extent of an object or set in a particular direction u, as shown in Figure 6-1. Such measurements arise in many ways in image reconstruction problems. In Chapter 4 we examined shadow based problems involving the silhouette of an object. A silhouette may be viewed as a set of support observations [8, 7], where the directions of observations u are confined to a particular subspace, as illustrated in Figure 6-2a. One-dimensional shadows or projections correspond precisely to a pair of support observations in opposite directions. In the realm of robotics, these support type measurements can arise from repeated grasps or probes by a gripper, as shown in Figure 6-2b. Finally, in low dose tomography the line integral observations may yield little more than shadow information [8], thus fitting into the silhouette framework above. Even when this is not the case, a preliminary step of projection support extraction coupled with object boundary estimation may be useful or desirable [9, 8]. This approach has proven particularly helpful in reflection tomography from laser range data [12]. These problems all share the common

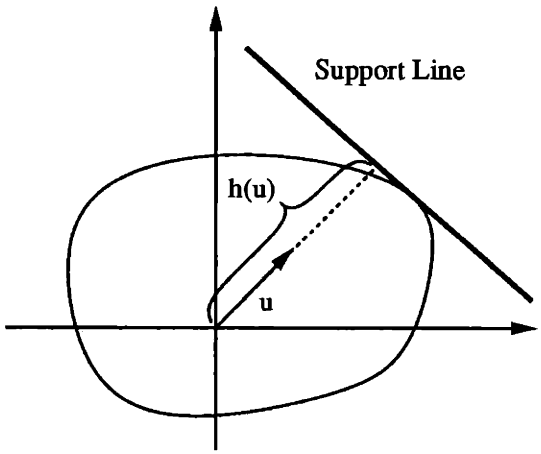

Figure 6-1: Illustration of a support measurement.

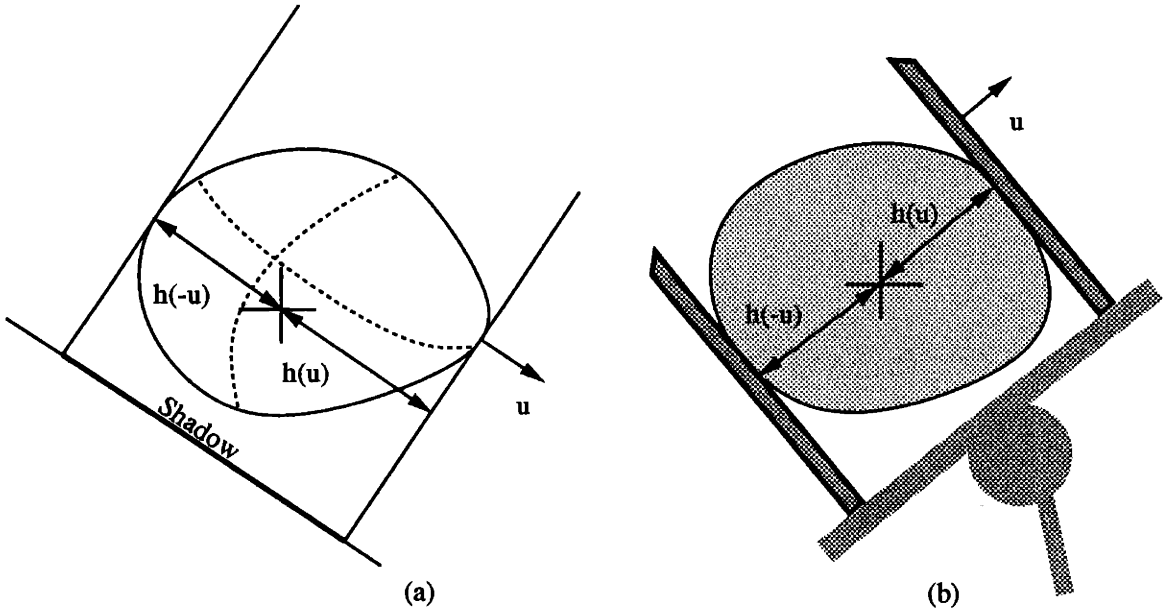

Figure 6-2: Applications of support measurements.

goal of *set reconstruction* from support measurements [87, 88, 89]. Besides being of general interest to computational geometers, set reconstruction from support data is also fundamental to robot vision [21] and chemical component analysis [89, 90, 91].

Outline

The work in the present chapter follows that of [9, 8]. The study of possibly noisy, and thus inconsistent, support measurements and the development of the underlying

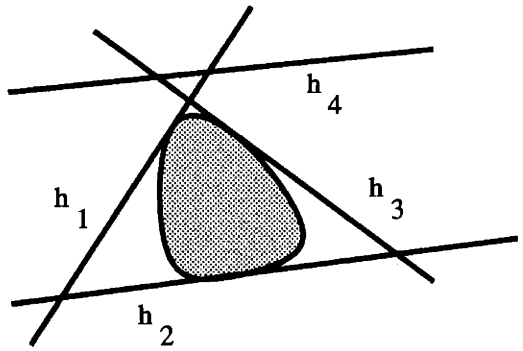

Figure 6-3: Illustration of inconsistent, noisy support measurements.

mathematics and geometry for use in constrained optimization problems is focused on. The continuous case of the support *function* and its properties is examined first. The conditions for a function to be a support function are reviewed in a general setting. An apparently little known such result for the planar case due to Rademacher and based on a determinantal inequality (which is also of a classical nature) is presented and a generalization developed.

After examining the continuous function case, the *discrete* problem arising from the sampling of a support function is treated. Such sampling appears because of the inherently discrete nature of support measurements in the applications under consideration. Due to the presence of noise, a group of such discrete observations may not, in general, be consistent, i.e. there might be no object that could have all the observations as support measurements. Such a situation is shown in Figure 6-3, where support measurements h_1, h_2, and h_3 are mutually consistent but h_4 is not. No object could have all these lines as support measurements. The possibility of such inconsistent observations leads to the examination of what constraints are required on a discrete set of support observations for consistency. The explicit statement of such constraints allows their use in estimation and optimization algorithms. For example, we might wish to find the set of consistent support values that is in some sense closest to the given observations. This work is an extension and generalization of the approach taken in [8], where the planar case was treated.

In addition to the above required constraints, we may wish to add additional constraints to regularize a problem or improve its qualitatively properties. An important constraint is smoothness. Intuitively, smoothness is related to the curvature of the object. A close connection exists between the constraints for validity of support

measurements and the curvature of a surface. Our insights and knowledge of these constraints are used to develop support based curvature notions. Since our measurements are necessarily discrete, our goal is to directly develop discrete versions of these quantities. Finally, in contrast to this fundamentally local approach, we develop global measures of smoothness based of the classical isoperimetric inequality.

6.2 Support Functions: The Continuous Case

In this section the properties of support *functions* are examined. The conditions for a function to be a support function are reviewed in a general setting, with a presentation of classical results and formal definitions. An apparently little known result due to Rademacher for the planar case is given. This result, based on a determinantal inequality, is interpreted geometrically as a *global* set of tests for consistency. An extension of this planar result to the general dimensional case is presented. The implications of such a set of tests for both planar and 3-dimensional objects is examined. These classically based results and our interpretation of them is used in the sequel to guide our examination, interpretation, and treatment of both the planar and three and higher dimensional conditions for discrete support *sample* consistency.

6.2.1 Support Function Definition and Properties

As discussed in the introduction, the (reduced) support function of an object may be thought of as the measure of extent of the object in a particular direction, as shown in Figure 6-1. For a given set, it is thus a scalar function of direction and hence a map from R^n to R. Formally, the support function of an object is given by the following definition:

Definition 4 (Support Function) *The* **support function** $H(v)$ *of an object* $\mathcal{O} \subset R^n$ *is given by:*

$$H(v) \equiv \sup_{x \in \mathcal{O}} x^T v$$

where the vector $v \in R^n$. *The* **reduced support function** $h(v)$ *is given by*

$$h(v) \equiv H(v/\|v\|)$$

and represents the distance from the origin of the corresponding supporting hyperplane to $\mathcal{O}$ in direction v.

Thus we see that it is actually the *reduced* support function that we obtain from physical measurements. Note that $H(v)$ is easily obtained from $h(v)$ due to the positive homogeneity of $H(v)$ to be discussed shortly ($H(\lambda v) = \lambda H(v)$ where $\lambda > 0$). In fact, the support function is determined by its values on the unit sphere $\|v\| = 1$.

Relation to Object Boundary

From the definition of the support function, every point x of $\mathcal{O}$ must satisfy the inequalities

$$x^T v \leq H(v) \quad \forall v$$

which together define the *convex hull* of the object. This relationship follows from the fact that the object $\mathcal{O}$ must be contained in the intersection of the support halfspaces $\mathcal{HS}(-u, -h(u))$, u a unit vector. Thus the (reduced) support function forms a complete representation of the convex hull of an object, and if the object is convex, then of the object itself. Since only the convex hull of an object may be recovered from support information, we really lose no generality in restricting our attention to convex objects.

A particularly simple relationship exists between the points on the boundary of a *strictly* convex object $\mathcal{O}$ (one where each support hyperplane has only one point in common with $\mathcal{O}$) and its support function $H(v)$. In this case, the support function $H(v)$ will be differentiable and the coordinates of the point p on the boundary of $\mathcal{O}$ at which the support hyperplane with exterior normal v is tangent to the object are given by [92, 93, 94]

$$p_i = \frac{\partial H(v)}{\partial v_i}.$$

Here p_i is the i-th coordinate of the boundary point and v_i is the i-th coordinate of the vector in the direction of the surface normal at the point. This formula may be rewritten more compactly as $p = \nabla H(v)$. It should be noted that $H(v)$ can be differentiable even if $\mathcal{O}$ is not smooth [46]. An example of such a case is given in Figure 6-4, where the object is strictly convex but not smooth at the point p.

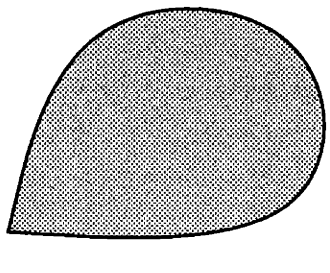

Figure 6-4: Illustration of non-smooth convex object with differentiable support function.

6.2.2 Relation to Curvature

Not surprisingly, the support function of a convex object has a close relationship to the object's curvature, see e.g. [92, 27]. Such results are collected here for future use in definitions of *discrete* curvature based on support samples. Some notation is needed to proceed. Suppose the second partial derivatives of the support function $H(v)$ of an object $\mathcal{O}$ exist on the unit ball. Let $[H_{ij}(v)]$ denote the Hessian matrix [38] of second partial derivatives of $H(v)$ ($H_{ij}(v) = \partial^2 H(v)/\partial v_i \partial v_j$). Further, let R_k, $k = 1, \ldots, (n-1)$ be the principal radii of curvature [71] of the boundary of $\mathcal{O}$ at the point $p(N)$ with unit outward normal N. In Chapter 4 the curvature of a surface was represented by the Hessian matrix $\mathsf{H}_{\mathcal{O}}(p)$ of a *height* function in some local coordinate system. The eigenvalues of this matrix are the principal curvatures of the surface at p and their reciprocals are the principal radii of curvature.

With these definitions, the principal radii of curvature at $p(N)$ satisfy the equation $|[H_{ij}(N)] - R_k I| = 0$, i.e. they are also eigenvalues of the support based Hessian matrix $[H_{ij}(N)]$. From homogeneity, $R = 0$ is always a trivial root of this equation so it is the remaining $(n-1)$ roots that are of interest. Equivalently to the above, the R_k are roots of the polynomial

$$R^{n-1} - D_1(H(N))R^{n-2} + D_2(H(N))R^{n-3} + \cdots + (-1)^{n-1} D_{n-1}(H(N))$$

where $D_\ell(H(v))$ denotes the sum of all principal minors of order ℓ of the matrix $[H_{ij}(v)]$. From this equation it follows that $D_\ell(H(v))$ is the ℓ-th elementary symmetric function of the R_k. In particular, we have that

$$D_{n-1}(H) = R_1 R_2 \ldots R_{n-1} = 1/\mathcal{K}(p)$$

where $\mathcal{K}(p)$ is the Gaussian curvature of $\mathcal{O}$ at p. Also:

$$D_1(H(N)) = H_{11}(N) + \cdots + H_{nn}(N) = R_1 + \cdots + R_{n-1}$$

where $R_1 + \cdots + R_{n-1}$ is the reciprocal of $(n-1)$ times the mean curvature at p. Note that for the strictly convex objects of concern here p identifies a unique N and vice versa. The implications and specializations for the 2-dimensional case are examined next.

The 2-Dimensional Case

For the 2-dimensional case $(n-1) = 1$ and we have that

$$D_{n-1}(H) = D_1(H) = H_{11}(N) + H_{22}(N) = 1/\mathcal{K}(p) \tag{6.1}$$

which is the reciprocal of the curvature of $\mathcal{O}$ at p. Recall that this relation only holds at points of $\mathcal{O}$ where the second derivatives of $H(v)$ exist. In the planar case $D_{n-1}(H)$ and $D_1(H)$ are the same since the Gaussian and mean curvatures are the same.

We may obtain an equivalent, perhaps more familiar, expression by parameterizing the unit normal N by angle θ. Making this substitution above yields, in terms of θ,

$$1/\mathcal{K}(\theta) = h_{\theta\theta}(\theta) + h(\theta) \tag{6.2}$$

where $h_{\theta\theta}$ is the second derivative of the (reduced) support function with respect to θ, the normal orientation. For example, for an ellipse oriented along the coordinate axes with semi-axis lengths a and b, the support function is given by

$$h(\theta) = \sqrt{a^2 \cos^2(\theta) + b^2 \sin^2(\theta)}$$

Applying (6.2) yields $\mathcal{K}(\theta) = h^3(\theta)/a^2b^2$. Interestingly, since the area of the ellipse is equal to πab, this expression is inversely proportional to the area squared.

Since the curvature of the surface of a convex object must always be positive it is clear that a necessary condition for a function to be a support function is that (6.2) be positive for all values of the argument and that the function be periodic. It can

also be proved that these conditions are sufficient for a function $h(\theta)$ or $H(v)$ (with unit v) to be the support function of a plane convex object [95]. Note that these conditions are *not* completely general since they require that the underlying function possess a second derivative. We shall have more to say about the requirements on a support function later.

The 3-Dimensional Case

Now we examine the curvature relationships for the 3-dimensional case. For D_1, the sum of the principle radii, we obtain:

$$D_1(H) = H_{11} + H_{22} + H_{33} = R_1 + R_2$$

where, recall, R_k are the principal radii of curvature of $\mathcal{O}$. Now, for the 3-dimensional case $(n-1) = 2$ so D_2 is the product of the principle radii and is given by:

$$\begin{aligned} D_2(H) &= \begin{vmatrix} H_{11}(N) & H_{12}(N) \\ H_{21}(N) & H_{22}(N) \end{vmatrix} + \begin{vmatrix} H_{22}(N) & H_{23}(N) \\ H_{23}(N) & H_{33}(N) \end{vmatrix} + \begin{vmatrix} H_{11}(N) & H_{13}(N) \\ H_{31}(N) & H_{33}(N) \end{vmatrix} \\ &= R_1 R_2 = 1/\mathcal{K}(N) \end{aligned}$$

where, as before, $H_{ij}(v)$ denotes the second partial derivative of $H(v)$ with respect to the elements v_i and v_j of its argument v. This expression yields the inverse of the Gaussian curvature of $\mathcal{O}$ at p. As an example, consider the ellipsoid oriented along the coordinate axes with semi-axes of length a, b, and, c. The support function $H(N)$ of the ellipsoid is given by

$$H(N) = \sqrt{N^T \begin{bmatrix} a^2 & 0 & 0 \\ 0 & b^2 & 0 \\ 0 & 0 & c^2 \end{bmatrix} N}.$$

Applying the formula for $D_2(H)$ yields the result that $\mathcal{K}(N) = H^4(N)/a^2b^2c^2$.

Similarly to the planar case, we may parameterize the set of unit normals N in R^3 by the two spherical coordinates angles θ and ϕ, as shown in Figure 6-5. Making this substitution in $D_2(H)$ above yields an expression analogous to (6.2) and which we have not seen before for the inverse of the Gaussian curvature as a function of the

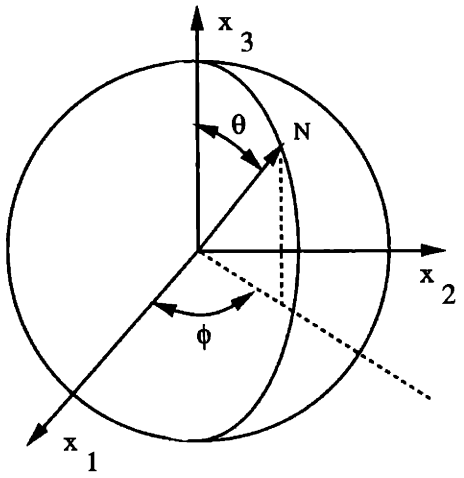

Figure 6-5: Unit normal parameterization.

normal coordinates (θ, ϕ):

$$1/\mathcal{K}(\theta,\phi) = (h + h_{\theta\theta})\left(h_{\phi\phi}\csc^2(\theta) + h_\theta \cot(\theta) + h\right) - \csc^2(\theta)\left(h_{\theta\phi} - h_\phi \cot(\theta)\right)^2$$

where the reduced support function $h(\theta, \phi)$ now depends on θ and ϕ and subscripts indicate partial differentiation. Similarly, for twice the mean curvature we obtain:

$$R_1 + R_2 = 2h + h_{\phi\phi}\csc^2(\theta) + h_\theta \cot(\theta) + h_{\theta\theta}$$

These continuous expressions will be of use in our development of discrete definitions of curvature in Section 6.4.

6.2.3 Relation to Area and Volume

Classically, the previous relationships between the support function and curvature of an object were used in formulas pertaining to area and volume [27]. Some of these relationships between an object's surface area and volume are now collected here. We have a special interest in the 2- and 3-dimensional cases. Discrete versions of these formulas will be used later in the development of discrete, global notions of object smoothness.

In general, the surface measure (area) of a convex object $\mathcal{O}$ is given by

$$\text{Surface Area} = \int_\Omega \frac{1}{\mathcal{K}}\, d\omega = \frac{1}{\pi}\int_\Omega \varsigma\, d\omega$$

where $\mathcal{K}$ is the Gaussian curvature of the object, Ω denotes the unit sphere, $d\omega$ is the corresponding area element, and ς is the cross sectional measure (area) of the projection of $\mathcal{O}$ onto the hyperplane through the origin with specified normal. The last equality above is Cauchy's formula for the area of the surface $\mathcal{O}$ as the average $< \varsigma >$ of the projections of $\mathcal{O}$.

For the n-dimensional volume of $\mathcal{O}$ we have [27, 94]:

$$\text{Volume} = \frac{1}{n}\int_\Omega \frac{H}{\mathcal{K}}\, d\omega \tag{6.3}$$

where $\mathcal{K}(N) = 1/D_{n-1}(H)$ is the Gaussian curvature.

The 2-Dimensional case

The planar case is examined next. The following formulas appear in Minkowski's work [27]. We use the reduced support function $h(\theta)$ parameterized by the normal angle θ. By specializing to a strictly convex object in the plane, one can show that:

$$\text{Perimeter} = \int_0^{2\pi} h(\theta)\, d\theta = \frac{1}{2}\int_0^{2\pi} \left(h(\theta) + h(-\theta)\right)\, d\theta = \pi < \mathsf{B} > \tag{6.4}$$

where $< \mathsf{B} >$ is the average breadth or 1-dimensional projection of the object. By substituting into the general equation for volume, one obtains the planar area enclosed by the convex object:

$$\text{Area} = \frac{1}{2}\int_0^{2\pi} h\left(h + h_{\theta\theta}\right)\, d\theta = \frac{1}{2}\int_0^{2\pi} \left(h^2 - h_\theta^2\right)\, d\theta \tag{6.5}$$

where subscripts denote differentiation with respect to the argument. The last equality follow from an application of Green's Theorem [27].

The 3-Dimensional case

In the 3-dimensional case one obtains for the surface area (see e.g. [27, 92])

$$\text{Area} = \int_{\Omega} \left(h^2 - \frac{1}{2}(h_\phi^2 \csc^2(\theta) + h_\theta^2) \right) d\omega$$

where Ω is the unit sphere, and $d\omega$ is the corresponding area element.

For the 3-dimensional case, we may substitute the expression for $\mathcal{K}$ in terms of the polar coordinates θ and ϕ presented earlier, obtaining:

$$\text{Volume} = \frac{1}{3}\int_{\Omega} h(h + h_{\theta\theta}) \left(\csc^2(\theta) h_{\phi\phi} + h_\theta \cot(\theta) + h \right) - h \csc^2(\theta) \left(h_{\theta\phi} - h_\phi \cot(\theta) \right)^2 \, d\omega$$

6.2.4 Characterization of Support Functions

A natural question is which functions $H(v)$ could be support functions of a convex body. Indeed, the problem is classical and the answer is provided by the following result, again classical:

Result 27 (Support Function Conditions) *A function $H(v)$ is the support function of a convex object if and only if it is defined for all vectors v and has the following properties:*

1. $H(0) = 0$.

2. $H(\alpha v) = \alpha H(v)$ *for* $\alpha > 0$.

3. $H(v + w) \leq H(v) + H(w)$, $\forall v, w \in R^n$.

A proof was given by Minkowski for the 3-dimensional case with other refinements provided by Rademacher and others (see e.g. [27]). Thus, only positively homogeneous, convex functions are support functions and vice versa. It is condition 3 of subadditivity that is the interesting one, as we shall see later. Note that these conditions are *global*, in the sense that they must hold for *all* vectors v and w and thus involve values of the support function over its entire range. In Result 28 we show that it is possible to reduce condition 3 to a series of *local* tests.

Determinantal Condition for the Planar Case

Rademacher has shown that it is possible in the 2-dimensional case to replace the subadditivity condition 3 of Result 27 by a determinantal condition on $H(v)$ over *unit* vectors. In particular, he showed that under conditions 1 and 2 of Result 27, condition 3 holds if and only if

$$\begin{vmatrix} H(u_1) & u_1^T \\ H(u_2) & u_2^T \\ H(u_3) & u_3^T \end{vmatrix} \begin{vmatrix} 1 & u_1^T \\ 1 & u_2^T \\ 1 & u_3^T \end{vmatrix} \geq 0$$

for all *unit* vectors u_1, u_2, and u_3 [27, 96]. Note that there is no requirement on the differentiability of $H(v)$. Using the definition of the *reduced* support function $h(v)$, this condition is also equivalent to:

$$\begin{vmatrix} h(u_1) & u_1^T \\ h(u_2) & u_2^T \\ h(u_3) & u_3^T \end{vmatrix} \begin{vmatrix} 1 & u_1^T \\ 1 & u_2^T \\ 1 & u_3^T \end{vmatrix} \geq 0 \tag{6.6}$$

for all *unit* vectors u_1, u_2, and u_3. Thus we now have a condition directly in terms of *physically measured* quantities. This condition is of interest for its interpretation. Using a determinantal equality, (6.6) may be rewritten as

$$\beta(u_i)\left(u_3^T \begin{bmatrix} 1 & u_1^T \\ 1 & u_2^T \end{bmatrix}^{-1} \begin{bmatrix} h(u_1) \\ h(u_2) \end{bmatrix} - h(u_3)\right) \geq 0 \tag{6.7}$$

where $\beta(u_i)$ can be shown to be a non-negative scalar that depends on the u_i. The term in parentheses in (6.7), which we shall call ρ, is the signed distance from the support line with normal u_3 to the intersection of the support lines with normals u_1 and u_2, see Figure 6-6.

In the plane then, the determinantal condition (6.7), and thus condition 3 of Result 27, requires that support functions satisfy an intuitive notion of consistency (as illustrated in Figure 6-3) for *all* triples of samples of the function. This intuition provides a geometric condition for a function to be a support function in the plane, but it is still a *global* condition, in the sense that all possible combinations of samples must be checked. An equivalent *local* result is given in Result 30.

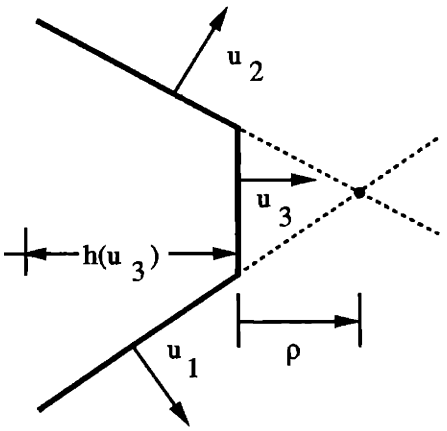

Figure 6-6: Illustration of 2-dimensional determinantal condition.

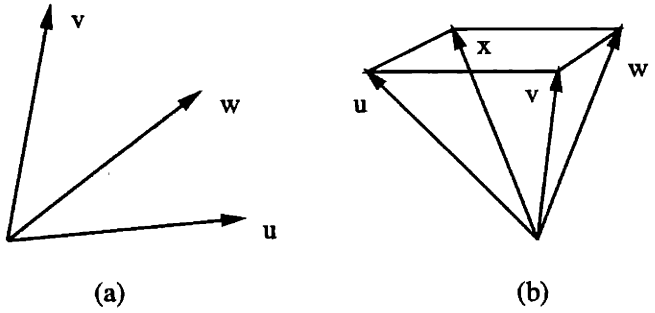

Figure 6-7: Difference between 2- and 3-dimensional situation.

Higher Dimensions

We now turn our attention to finding an equivalent of the determinantal inequality condition (6.6) for the higher dimensional case. Unfortunately, in three and higher dimensions the exactly analogous condition (i.e. validity of such a determinant inequality for *all* vectors u_i) has been shown by Rademacher to be satisfied only by the support functions of balls [96]. We identify the difficulty in directly extending this result and present a natural generalization of condition (6.6) that is valid for all dimensions. The result appears to be new.

The difference between the two and higher dimensional cases is that in the plane, given three vectors, one vector can *always* be written as the convex combination of the other two, as shown in Figure 6-7a where w is in the positive cone of u and v. In higher dimensions this is not necessarily true, as illustrated by the combination of normals in (b) of the same figure. If we consider the geometric interpretation of

the test, it seems reasonable to suppose that by restricting our attention to groups of vectors for which this cone condition *is* satisfied, we might obtain the desired result. This is precisely what we do, yielding the following new result:

Result 28 (General Inequality Condition) *A function $H(v)$ is the support function of a convex object if and only if it is defined for all v and has the following properties:*

1′. $H(0) = 0$.

2′. $H(\alpha v) = \alpha H(v)$ *for* $\alpha > 0$.

3′. *The following determinantal inequality is satisfied for all $(n+1)$-tuples of unit vectors u_i with one in the full positive cone of the others:*

$$\left| \begin{array}{cc} H(u_1) & u_1^T \\ H(u_2) & u_2^T \\ \vdots & \vdots \\ H(u_{n+1}) & u_{n+1}^T \end{array} \right| \left| \begin{array}{cc} 1 & u_1^T \\ 1 & u_2^T \\ \vdots & \vdots \\ 1 & u_{n+1}^T \end{array} \right| \geq 0$$

In a finite dimensional space a cone is said to be *full* if it cannot be contained in a proper subspace. The proof of the result is in Appendix 6-A. As before, there is no requirement on the differentiability of $H(v)$.

Since condition 3′ of Result 28 uses only unit vectors, it is equivalent to the following condition on the *reduced* support function of $H(v)$:

$$\left| \begin{array}{cc} h(u_1) & u_1^T \\ h(u_2) & u_2^T \\ \vdots & \vdots \\ h(u_{n+1}) & u_{n+1}^T \end{array} \right| \left| \begin{array}{cc} 1 & u_1^T \\ 1 & u_2^T \\ \vdots & \vdots \\ 1 & u_{n+1}^T \end{array} \right| \geq 0 \tag{6.8}$$

for all unit vectors u_i with one in the full positive cone of the others. Note that our condition also recovers the planar case. Assume u_{n+1} is in the positive cone of the remaining $\{u_i\}$. By using the same determinantal equality as before, we may rewrite (6.8) as

$$\beta(u_i) \left(u_{n+1}^T \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_n^T \end{bmatrix}^{-1} \begin{bmatrix} h(u_1) \\ h(u_2) \\ \vdots \\ h(u_n) \end{bmatrix} - h(u_{n+1}) \right) \geq 0$$

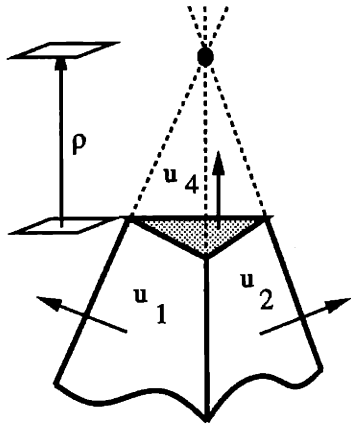

Figure 6-8: Illustration of 3-dimensional determinantal condition.

where again the scalar function $\beta(u_i)$ is non-negative. The second term may be naturally interpreted as the signed distance ρ, positive in the direction of u_{n+1}, from the support hyperplane with normal u_{n+1} to the point determined by the intersection of the hyperplanes with normals given by u_i, $i = 1, \ldots, n$, as shown for the $n = 3$ case in Figure 6-8.

Condition 3′ of our Result 28 thus generalizes our intuition of the planar case to arbitrary dimensions. As in the planar case, this condition is still a global one, in the sense that *all* $(n + 1)$-tuples of vectors satisfying a positive cone condition must be checked. In the following sections Result 28 is used to develop a connection between the conditions that characterize a support function and the consistency of a given set of support samples. This section concludes our examination of the classical support *function* results. Next we turn our attention to the discrete case of a *sampled* support function.

6.3 Consistency of Support Samples

In this section the discrete case arising from the sampling of a support function is focused on. Due to the presence of noise, a group of such discrete observations will not, in general, be consistent, i.e. there might be no object that could have all the observations as support measurements. To be precise, we term a set of support samples *consistent* if there exists a valid support function whose values at the sample points match the given set. An example of such a situation was given in Figure 6-3.

As a result of such potential difficulties with noisy support measurements, our main interest is solving the following problem, phrased in terms of the reduced support function:

Problem 9 (Support Consistency) *Given a set of samples* $\{h_i\}$ *in directions* $\{u_i\}$, *determine if there exists a valid support function* $h(v)$ *such that* $h(u_i) = h_i$.

One obvious approach we could take is to attempt to explicitly find offending hyperplanes, such as h_4 in Figure 6-3. We show that this problem is equivalent to finding non-binding constraints in a linear programming (LP) problem. It turns out that this task is computationally expensive, essentially necessitating the solution of a dual LP problem itself. Tractable *probabilistic* approaches exist however, and are presented. While partially overcoming the problem, these approaches are unattractive for use in constrained optimization.

As a result, the explicit search for the offending support values is abandoned, and instead tests or constraints are developed which simply tell of the *existence* of inconsistency. This approach is actually preferable since it does not attempt to assign the inconsistency to any particular measurement. Different optimizations are free to use the conditions as a constraint in the reconstruction of a consistent set as they see fit. Such a constraint was developed for the case of planar objects with support line observations at equal angles in [8]. This work on the 2-dimensional problem is review and linked with our classical results on support functions. The test in [8] is reinterpreted as a series of *local* constraints on the support observations. The notion of the locality for this planar case is trivially imposed by the monotonic ordering of the corresponding support line angles.

The previous 2-dimensional development is then extended to observations at arbitrary angles and in arbitrary dimensions, resulting from the use of support *hyperplane* samples. A corresponding local consistency constraint for the general case involving the support hyperplane measurements is developed. Defining the notion of locality of the support hyperplanes is more complicated for the general case, as there is no natural way of ordering the higher dimensional support sample observations as there was for the 2-dimensional case. The key is a certain generalization of the notion of nearest neighbor utilizing a simple positive cone condition on the respective support sample normals.

6.3.1 Finding Non-Binding Constraints

Certainly one approach we could take to testing for consistency of a set of support values $\{h_i(u_i)\}$ is to attempt to explicitly find those support hyperplanes which are inconsistent. This idea is illustrated in Figure 6-3, where h_1, h_2, and h_3 are consistent with the shown object (and thus with the corresponding support function of the object), while h_4 is inconsistent with any object. Note that we implicitly have to know what the "interior" of the set is, i.e. where the object is supposed to be. We assume that our support observations are directed, in the sense that we know which side of them the underlying object is on.

Direct Computation

Essentially what we are doing when we say that h_4 is inconsistent is implicitly intersecting the directed halfspaces provided by the h_i to obtain a convex polyhedral region and then attempting to identify those hyperplanes that do not contribute to this region. This problem arises in connection with linear programming (LP). In the LP context the intersection region described above corresponds to the feasible region and the hyperplanes arise from the linear inequality constraints of the optimization problem. The desire is to eliminate redundancies in the constraint set. Such redundancies appear in the form of inequalities which can never be binding. The motivation in the LP case is to reduce the computational burden in solving a problem and the potential benefits are enough that some form of (suboptimal) preprocessing is often done in practice. Even then, the study of such problems from the point of view of the individual constraints is relatively young, starting in the early sixties [97, 98].

Much of the work in the area is devoted to characterizations of redundant and non-redundant constraining hyperplanes (consistent and inconsistent in our framework). The following lemma is one of the original such examples (see e.g. [99]). We phrase it in our framework.

Lemma 7 (Turnover Lemma [98]) *The support sample pair (u_k, h_k) is inconsistent in the set $\{(u_i, h_i)\}$ if and only if the following system of inequalities has no solution*

$$\begin{aligned} u_i^T x &\leq h_i, \quad i \neq k \\ u_k^T x &> h_k \end{aligned}$$

This lemma shows that a redundant constraint is related to the feasibility of an LP problem. Other results along these lines attempt to identify which variables in an LP problem are "permanently basic," (that is, are retained in the simplex tableau and hence binding [100]) or to solve an equivalent dual problem [101].

Unfortunately, using such characterizations to explicitly find redundant bounding hyperplanes in practice demands the solution of other related LP problems. In fact, the testing of each hyperplane requires the solution of a separate LP problem [99], in general $(n + 1)$ LP problems need to be solved to check all the hyperplanes. Unfortunately, as stated in [99, pg. 69] for the LP problem, "there are no deterministic methods to identify nonbinding constraints without finding the optimal solution to the problem." In short, the associated computational burden is large enough that other, heuristic approaches are commonly used as preprocessors for LP problems.

Hit-and-Run Approaches

As an alternative to exhaustively identifying the redundant constraints these methods work by attempting to identify the *non*-redundant bounding hyperplanes. These approaches are probabilistic, in the sense that they attempt to identify all the active constraints *in the limit* of some process [99]. These approaches work as follows. A random set of interior points is found and a search is carried out from each of these points in a random direction. The constraint first encountered in the random direction (and its opposite) are identified as being nonredundant and a new point interior point is generated. The different methods mainly differ in how they generate the sequence of random directions. For example, in the hypersphere directions method, the directions are generated from a uniform distribution on the hypersphere, while in the coordinate directions method they are chosen uniformly from the coordinate direction vectors and their negatives. In both these cases, the next interior point is chosen randomly along the line connecting the current boundary intersection points. These methods are referred to as *hit-and-run algorithms*, see e.g. [99, 102].

It has been shown that both the above mentioned algorithms generate a sequence of points that has a limiting distribution which is uniform over the interior of the underlying polytope. These results imply that both of these methods identify each nonredundant constraint with probability one as the number of tests goes to infinity. The utility of these approaches is the speed with which they may be implemented. An important question, of course, is when to terminate the procedure, since in general the true number of binding constraints is not known apriori. In [102] an optimal

Bayesian stopping criterion for these type of approaches is found. The algorithms may be terminated when the optimal Bayesian estimate of the number of undiscovered nonredundant constraints is 0. Such an approach appears to work well in practice.

Despite their interest and obvious relation to the issues of this chapter, there are several problems in applying the approaches of this section in our setting. Firstly, the redundant hyperplanes are only implicitly (since the hit-and-run methods find the *non-redundant* ones) or numerically identified. This means that we cannot conveniently incorporate such conditions into a constrained reconstruction setting. There is also a fundamental bias in the orientation of attempting to find inconsistent hyperplanes, in that all the error is assumed to lie in the inconsistent support measurements, with the other, "consistent" measurements assumed perfect [8]. It is much easier just to test for consistency of the *entire set* rather than to try assigning the inconsistency to a particular measurement. Next we consider such tests, suitable for optimization and straightforward to implement. The results presented next may even prove useful in the original LP problem we considered here.

6.3.2 Identifying Consistency

Here attention is turned to overall tests for consistency of a set of support samples solving Problem 9. Such a test for consistency actually exists in Result 28. We need only apply it to the given support samples h_i (with a certain positive cone constraint) to see if they are consistent. Such an application yields the following corollary of Result 28.

Corollary 6 (Discrete Consistency) *Given a set of support samples h_i with associated unit normals $u_i \in R^n$, suppose that each of the u_i is contained in the full positive cone of at least one n-tuple of the remaining normals. In this case, the set of samples is consistent if and only if condition 3′ of Result 28 is satisfied by the samples of the set.*

If there is a valid support function with the given samples then these samples will certainly satisfy condition 3′. Conversely, if the samples satisfy condition 3′ then the polyhedron obtained by intersecting the corresponding halfspaces yields a valid support function whose samples match with the given samples.

The hypotheses of the corollary may even be weakened a little to extend the discrete result to a wider class of problems. Since in the discrete case we do not

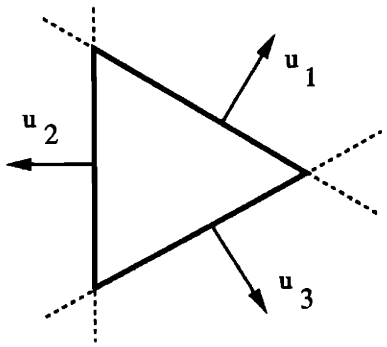

Figure 6-9: Illustration of the necessity of an addition condition.

have access to samples at all orientations, to produce a consistency test we need a condition to ensure that the consistency of each sample is checked. Such a condition was provided in the corollary by the requirement that every normal be in the full positive cone of at least one n-tuple of the remaining normals. In this way the positive cone test of 3′ is assured of checking all the samples. One way of viewing such a condition is that we require that support samples be taken with a certain minimum of uniformity and number. For the planar case of [8] such a constraint was effectively imposed by having at least three samples every π radians, for a minimum of 5 samples total. A situation *not* satisfying this condition, which might certainly arise in practice, is shown in Figure 6-9, where *none* of the u_i is in the positive cone of the others. Since none of these normals is in the positive cone of a set, there is nothing to which we may apply the determinant test 3′ of Result 28!

The problem is that, in reality, to mimic Result 27 we should be checking *all* cone inclusions, both positive and negative. Because of the convexity of the underlying object, however, when the support sample normals are close enough together checking only the positive cone cases suffices, as shown by the corollary. In particular, this is the situation for the continuous support function condition of Result 28. As a consequence, instead of constraining the sampling, as discussed above, we could expand the test over both positive *and* negative cone inclusions. In fact, to reduce the number of tests we could use the following compromise. First, all samples with normals in the positive cone of some n-tuple would be checked as before. Then, if there were any remaining samples whose normals were *not* in the positive cone of some set, these samples would be checked with respect to all groups in which they were in the *negative cone*, with the inequality of the test in 3′ now reversed. Finally, if there were normals which were not in the positive or negative cone of any set, then testing

them would be irrelevant anyway (consider the planar case with two samples at right angles to each other). While these ideas provide a clean way to apply the discrete test to any situation (including that of simplices as pictured in the Figure 6-9), for most practical cases, where the original positive cone restriction is satisfied anyway, this process unnecessarily increases the complexity of the process. For this reason, in the present work we concentrate on the former approach of assuming that every normal is in the full positive cone of at least one n-tuple other normals.

While the inequality tests resulting from Corollary 6 are convenient, in that they are simple linear functions of the support measurements h_i, the procedure is problematic in that *all* positive cone combinations must be checked for consistency. The number of such tests grows combinatorially with the number of observations. For example, in the planar case with support values equally spaced in angle, if m is the number of observations then the number of tests grows as $m^2/8$. This growth becomes worse in higher dimensions because of the increased number of degrees of freedom. As a result we seek a *local* test, utilizing only local support information in its application. This approach may be viewed as a discrete version of the curvature constraint discussed in connection with (6.2) (though it does not require differentiability of the underlying support function).

Local Tests: The 2-Dimensional Case

In the plane we may parameterize unit vectors u_i by their angle θ so that $u_i = [\cos(\theta_i), \sin(\theta_i)]^T$. Suppose the θ_i are chosen so that $\theta_{i+2} - \theta_i < \pi/2$, with $\theta_{i+1} \geq \theta_i$. Applying Corollary 6 to this planar case, we obtain the following consistency condition for a set of support samples which must hold for *all* ordered triples $(\theta_i, \theta_j, \theta_k)$, where the normal associated with θ_j is in the positive cone of the other two normals:

$$[\sin(\Delta\theta_{kj}) - \sin(\Delta\theta_{kj} + \Delta\theta_{ji}) + \sin(\Delta\theta_{ji})] \times$$
$$[\ \sin(\Delta\theta_{kj}) \quad -\sin(\Delta\theta_{kj} + \Delta\theta_{ji}) \quad \sin(\Delta\theta_{ji})\ ] \begin{bmatrix} h_{i-1} \\ h_i \\ h_{i+1} \end{bmatrix} \geq 0 \qquad (6.9)$$

where $\Delta\theta_{ji} = \theta_j - \theta_i$ is the angular difference between normal j and normal i.

The test associated with (6.9) is not a local test since *all* triples must be checked, not just adjacent ones. Such a local test was given in [8] for the planar, equal-angle case (but its significance as a local test was not brought out by Prince). If $\mathbf{h} = [h_1, h_2 \cdots, h_n]^T$ is the vector of support samples (termed the support vector [8]),

then a constraint on $\mathbf{h}$ was given as $Q\mathbf{h} \geq 0$ where the matrix Q may be written as

$$Q = \begin{bmatrix} -1 & \sec(\Delta\theta)/2 & 0 & & \sec(\Delta\theta)/2 \\ \sec(\Delta\theta)/2 & -1 & \sec(\Delta\theta)/2 & \cdots \quad \cdots & 0 \\ 0 & \sec(\Delta\theta)/2 & -1 & & \vdots \\ \vdots & 0 & \sec(\Delta\theta)/2 & \cdots \quad \cdots & 0 \\ 0 & \vdots & & & \sec(\Delta\theta)/2 \\ \sec(\Delta\theta)/2 & 0 & 0 & & -1 \end{bmatrix} \tag{6.10}$$

and $\Delta\theta > 0$ is the uniform angle increment between sample normals. From the circulant structure of Q we may interpret this as a local test, since in each row only adjacent support samples are involved. In fact each row of $Q\mathbf{h} \geq 0$ is a test between neighboring samples of the form

$$\begin{bmatrix} \sin(\Delta\theta) & -\sin(2\Delta\theta) & \sin(\Delta\theta) \end{bmatrix} \begin{bmatrix} h_{i-1} \\ h_i \\ h_{i+1} \end{bmatrix} \geq 0.$$

Comparing this expression with the expression in (6.9) applied to the planar, equally spaced normal case (where the first term is always positive), we see that the result in [8] serves to *localize* the global condition of Result 28. Geometrically, this test is just the one presented in Figure 6-6, only applied locally (and for the equal angle case). Thus we have the observation that, in the plane, if a set of equal-angle samples are consistent locally (among adjacent triples) then they are consistent globally (over all groupings).

Note that in contrast to the approaches discussed in Section 6.3.1, this type of test does *not* identify which constraints are inconsistent. To see this, consider the situation shown in Figure 6-10, where the intersections of the adjacent support lines used in the local tests are shown as the points p_i and the object is assumed contained in the shaded region. The support measurements with normals u_2, u_3, u_4, would fail the local test at point p_2, since the line associated with u_3 is behind p_2 (in the direction given by u_3). While this failure does confirm the existence of inconsistency, note that the set of samples associated with u_1, u_2, and u_3, would pass their local test at p_1. The distance from the u_2 support line to p_1 is positive in the direction given by u_2 so the local test is satisfied. Thus, while the sample with normal u_2 is also inconsistent, it is not identified by the local tests.

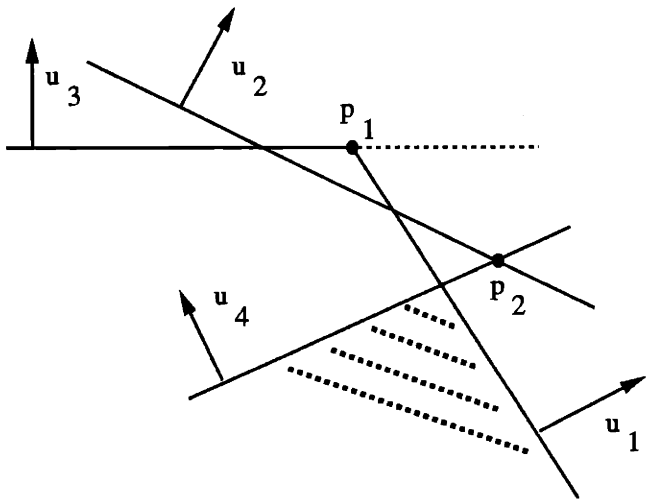

Figure 6-10: Illustration of non-specificity of local tests.

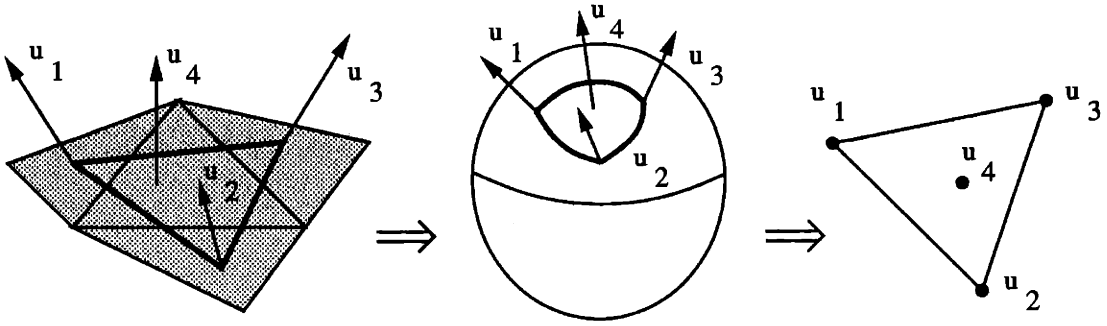

Figure 6-11: Graphical representation scheme for normal relationships.

6.3.3 Local Tests: The General Case

The above planar local test is now extended to the general case of arbitrary dimension and arbitrary normal direction. These extensions serve to unify the material on support consistency presented thus far. The key to this generalization is the observation from the planar case that the structure of the local and global tests is the same, the only difference being that one is applied locally and the other globally. The generalization thus reduces to showing that local consistency implies global consistency.

A result is first given that allows satisfaction of the determinant test (6.8) over given sub-domains of the Gaussian sphere to be extended to satisfaction over a larger domain. We term this result a *consistency merging* result. Before presenting the result we provide a geometrical description of it. To this end, consider the situation shown in Figure 6-11 for the 3-dimensional case. The normals to support planes are mapped to

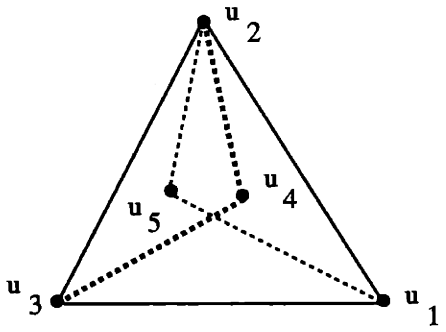

Figure 6-12: Illustration of meaning of Result 29.

points representing their tips on the Gaussian sphere, as shown. A spherical triangle connects these points on the Gaussian sphere. We represent this spherical triangle by a corresponding planar triangle. Any point in the positive cone of the vertex normals is a point in the triangle and vice versa. For example, in Figure 6-11 u_4 is in the positive cone of u_1, u_2, and u_3.

With this scheme, the result in three dimensions is illustrated in Figure 6-12. Here u_4 is in the positive cone of $\{u_1, u_2, u_5\}$ and conversely u_5 is in the positive cone of $\{u_2, u_3, u_4\}$. In three dimensions, the result states that, given the above inclusions, if $\{u_1, u_2, u_4, u_5\}$ form a consistent set and $\{u_2, u_3, u_4, u_5\}$ form a consistent set then $\{u_1, u_2, u_3, u_4\}$ also form a consistent set (and by symmetry so does $\{u_1, u_2, u_3, u_5\}$). Thus, consistency over the smaller triangles implies consistency over the larger triangle. In the higher dimensional cases the triangle of Figure 6-12 becomes an $(n-1)$-dimensional *simplex* and the interior triangles sub-simplices. The full result is as follows:

Result 29 (Consistency Merging) *Given a set of* $(n+2)$ *support samples* h_i *with associated unit normals* u_i*, suppose* $u_{n+1} \in \mathsf{cone}^+\{u_1, \ldots, u_{n-1}, u_{n+2}\}$ *and* $u_{n+2} \in \mathsf{cone}^+\{u_2, \ldots, u_n, u_{n+1}\}$*, and that both these cones are full. If both the sets of support samples* $\{h_1, \ldots, h_{n-1}, h_{n+1}, h_{n+2}\}$ *and* $\{h_2, \ldots, h_n, h_{n+1}, h_{n+2}\}$ *are consistent then so are the enlarged sets* $\{h_1, \ldots, h_n, h_{n+1}\}$ *and* $\{h_1, \ldots, h_n, h_{n+2}\}$.

In the above, cone^+ denotes the positive cone of a set and by consistency of a set we mean satisfaction of the determinantal inequality (6.8) by the set. The proof of this result is in Appendix 6-B. Notice that under the conditions of the result $u_{n+1} \in \mathsf{cone}^+\{u_1, \ldots, u_n\}$ and $u_{n+2} \in \mathsf{cone}^+\{u_1, \ldots, u_n\}$.

To proceed with the generalization of the planar local result, a suitable notion of "local" needs to be defined for the general case, or, in the context of Result 29,

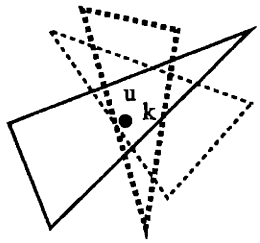

Figure 6-13: Illustration of a local family

we need to know the minimal domain over which consistency must be satisfied. In the planar case this definition of locality was straightforward, depending on normal ordering. In higher dimensions, however, the situation is not so clear. Adjacent faces do not necessarily correspond to nearest normals anymore. A natural notion of locality is suggested both by condition 3′ of Result 28 and by Result 29 with their emphasis on a positive cone condition on the unit normals. Given a normal, we define what we mean to be "local" to that normal in the following[1]:

Definition 5 (Local Family) *Given a set of m distinct vectors* $\{u_i\}$ *in* R^n *and a member from this set* u_k*, we define the* **local family** $\mathcal{LF}_{u_k}$ *with respect to this member to be the set of all distinct* $(n+1)$*-tuples of vectors* $\{u_j\}$ *from the original set with the property that* u_k *is one element and the remaining n vectors contain* u_k **and only** u_k *in their full positive cone.*

Thus, the local family corresponding to the member u_k is a *set of* $(n+1)$*-tuples*, containing u_k and with the property that the only element of the set contained in their positive cone is the generating element u_k. In terms of the paradigm of Figure 6-11, a local family is the set of all (spherical) triangles (simplices in higher dimensions) containing the given normal u_k but no others, as shown in Figure 6-13. In contrast to the planar case, where there was just a single local neighbor to test, this notion of locality implies a *family* of tests at each normal, one for each $(n+1)$-tuple in the local family.

Local Constraint

With these ideas of locality defined we are prepared to present a result generalizing and extending the previous planar condition showing that local consistency and global consistency are equivalent.

[1]This idea of a local *family* was suggested by Sanjeev Kulkarni.

Result 30 (Local Consistency $\Longleftrightarrow$ Global Consistency) *Given a set of support samples $\{h_i\}$ with associated unit normals $\{u_i\}$ in R^n, suppose that each element of the set u_k is in the full positive cone of at least one n-tuple of the remaining u_i (so no local family is empty). Then the overall set of samples is consistent if and only if for each normal u_k, all elements of the local family $\mathcal{LF}_{u_k}$ are consistent. In other words, the overall set is consistent if and only if all elements of all local families are consistent.*

Again, consistency of a set means satisfaction of (6.8). Thus, we have shown that a set is globally consistent if and only if it is locally consistent, where locality is defined in the sense of the local family of a normal. The proof of the result is in Appendix 6-C.

Note that each of the tests in Result 30 is linear in the support samples h_i. As a result, given a specific set of m samples, we may write the corresponding test as

$$\bar{Q}\mathbf{h} \geq 0 \tag{6.11}$$

where $\mathbf{h}$ is again the support vector composed of the stacked samples h_i and $\bar{Q}$ is an $m \times m$ sparse matrix guaranteed to have only $n+1$ non-zero entries in each row. The matrix $\bar{Q}$ is thus the generalization to higher dimensions of the matrix Q given for the planar case in (6.10). Since the definition of the local families depends only on the sample normals u_i and not on the support samples themselves, $\bar{Q}$ depends only on the normals u_i. Consequently, once these directions are fixed the matrix $\bar{Q}$ may be *precomputed* and applied to many different sets of measurements.

This form of constraint is particularly convenient for constrained support reconstruction. For example, suppose that we are given a set of noisy support observations in the vector y and that we wish to reconstruct the least square error estimate of $\mathbf{h}$ from these observations subject to consistency. The resulting problem combines (6.11) with a least squares criteria to yield the following linear inequality constrained least squares problem, which is straightforward to solve:

$$\arg\min_{\mathbf{h}\bar{Q}\geq 0} \|\mathbf{h} - y\|_2$$

This model of known u_i but noisy h_i is reasonable for many problems, particularly medical tomography problems, where the user may exercise great control over the orientation of the data acquisition. The situation is obviously more complicated if

we consider the, perhaps more realistic situation, of noisy measurements h_i coupled with imperfectly known geometry u_i. Such situations arise in geophysical problems and target tracking [12].

As an example of the savings involved in using a local test instead of the general global result of Corollary 6, we consider the situation presented by the icosahedron in 3 space. The icosahedron, with 20 faces, is the completely regular polyhedron with the largest number of equally spaced faces and thus normals. Applying Corollary 6 to this case with 20 normals would yield a total of 1620 tests of the type (6.8). In other words, the support vector would have 20 elements and the corresponding matrix $\bar{Q}$ of the matrix inequality would be 1620×20. In contrast, using the local test given by Result 30 results in only 320 inequality tests, or a 320×20 matrix $\bar{Q}$. This is still large, but over a factor of 5 better than before.

Identifying the local tests in practice is laborious but straightforward. For each sample normal u_k one may exhaustively test all possible remaining n-tuples to see if u_k is in the resulting positive cone. We may test if u_k is in the positive cone of a given n-tuple by checking the coefficients of the vector $[u_1|u_2|\cdots|u_n]^{-1}u_k$ for positivity, where the columns of the matrix $[u_1|u_2|\cdots|u_n]$ are composed of the vectors of the n-tuple. As discussed above, this need only be done once for a given set of sample directions.

6.4 Discrete Curvature Concepts

In this section, discrete concepts of surface curvature are examined. Our aim is to develop discrete measures of object smoothness for inclusion in a constrained polyhedral, support based reconstruction. One approach taken is to perform this analysis *locally*, by minimizing the maximum curvature value over the object surface. This operation produces smooth objects by preventing the points of high curvature associated with roughness. The close connection between support consistency and local surface curvature shown in section 6.2.2 is exploited to develop discrete, support based notions of surface curvature. The above smoothness constraint may then be phrased in terms of the minimum value of the *discrete* curvature allowed in the reconstructed object. Such an approach was taken for the planar case in [8]. Both this planar case as well as the 3-dimensional one are examined here.

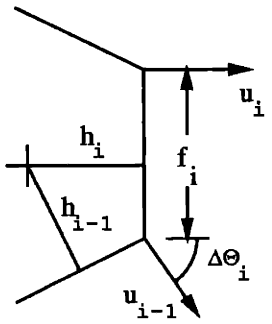

Figure 6-14: Polygonal object resulting from support samples.

6.4.1 Planar Case

In this section discrete notions of curvature for the planar case are examined. A definition of discrete curvature was presented in [8] based on the planar support consistency tests given earlier. Both this definition and some alternative ones are presented. Limiting analyses of these different curvature ideas are given which suggest the desirability of our formulations over the previous work. With this background, the 3-dimensional case is treated.

Equal Angles

The curvature $\mathcal{K}$ of the boundary of a planar object may be defined as the rate of change of the normal angle with respect to arc length along the boundary

$$\mathcal{K} \equiv \frac{d\theta}{df}$$

where θ is the angle of the boundary normal and f is the boundary measure of the object. In the discrete case, the set of support samples defines a polygonal region, formed by intersecting the directed support halfspaces. Consider the section of such a region shown in Figure 6-14. We can see that, roughly speaking, in going from u_{i-1} to u_i the normal has gone through an angle change of $\Delta\theta_i = \theta_i - \theta_{i-1}$ over the length of the face f_i. Using this reasoning, the i-th discrete curvature value $\mathcal{K}_1(i)$ was defined in [8] for the planar case as:

$$\mathcal{K}_1(i) \equiv \frac{\Delta\theta_i}{f_i}. \tag{6.12}$$

where the subscript 1 is used to distinguish this definition of discrete curvature from others to be proposed. If ρ_i is the i-th entry of the vector $Q\mathbf{h}$ that arises in the planar, equal-angle inequality test defined in Section 6.3.2, where Q was defined in (6.10), then one can show that

$$\mathcal{K}_1(i) = \frac{\Delta\theta \tan(\Delta\theta)}{2\rho_i}.$$

Geometrically, ρ_i was defined in Figure 6-6 as the distance from a support line to the intersection point of its neighbors. Thus, the curvature $\mathcal{K}_1(i)$ is related to the support vector $\mathbf{h}$. In fact, analysis in [103] shows that this definition of $\mathcal{K}_1(i)$ converges to the true planar curvature expression $(h_{\theta\theta} + h)^{-1}$ as $\Delta\theta$ goes to zero. In [8] only the equal angle case was considered. We now examine the properties of this curvature estimate under the more general condition of different $\Delta\theta_i$.

Convergence Analysis

To investigate the convergence properties of different definitions of curvature, they will be applied to a general parabola centered on the origin and aligned with the coordinate axes. Since the surface of any smooth, convex region may be locally modeled to second order as a parabola, limiting properties obtained on centered parabolas will match those of any smooth surface to second order. Since curvature is a second order property, no generality is lost by restricting our attention to such an analysis. This use of a second order Taylor representation of a surface as a quadratic function is the same technique we used in Chapter 4 for our curvature analysis. For the planar case the parabolic function $z = \frac{1}{2}gx^2$ is used, as illustrated in Figure 6-15 along with representative support samples. The curvature of this boundary at the origin (the location of the k-th sample and the assumed point of interest) is given by $\mathcal{K}_{\text{true}} = g$.

Limiting properties of $\mathcal{K}_1(i)$. First, the curvature definition $\mathcal{K}_1(i)$ of (6.12) is applied to the quadratic boundary at the origin, see Figure 6-16. In Appendix 6-D.1 we show that

$$\mathcal{K}_1(i) = \frac{2g\Delta\theta_i}{\tan(\Delta\theta_i) + \tan(\Delta\theta_{i+1})}$$

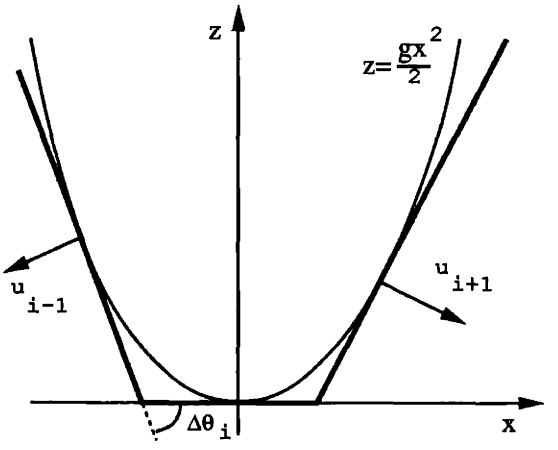

Figure 6-15: Generic parabola for curvature analysis.

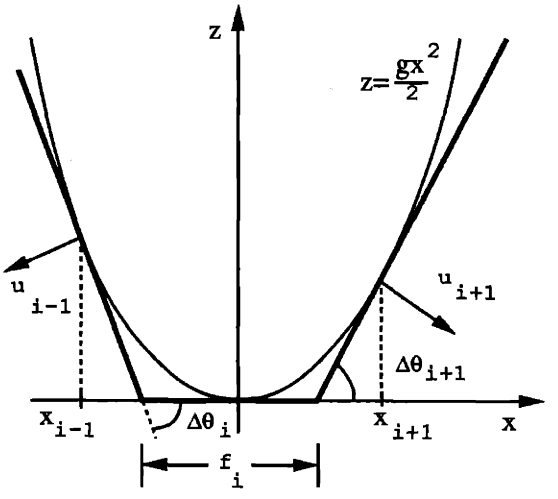

Figure 6-16: Definition 1 of discrete curvature, $\mathcal{K}_1$.

Let $\Delta\theta_{i+1} = \alpha\Delta\theta_i$ so that the scalar α measures the relative rate of convergence of the two angles. Now in the limit as $\Delta\theta_i$ goes to zero we obtain

$$\lim_{\Delta\theta_i \to 0} \mathcal{K}_1(i) = 2g\frac{1}{1+\alpha}.$$

Thus the correct curvature is obtained only in the case that $\alpha = 1$, that is, when the convergence rates are equal. In particular, this equal convergence rate is true for the equal angle case investigated in [103]. In general, however, unless we can assure this equality in the limit, the definition $\mathcal{K}_1(i)$ will give erroneous results.

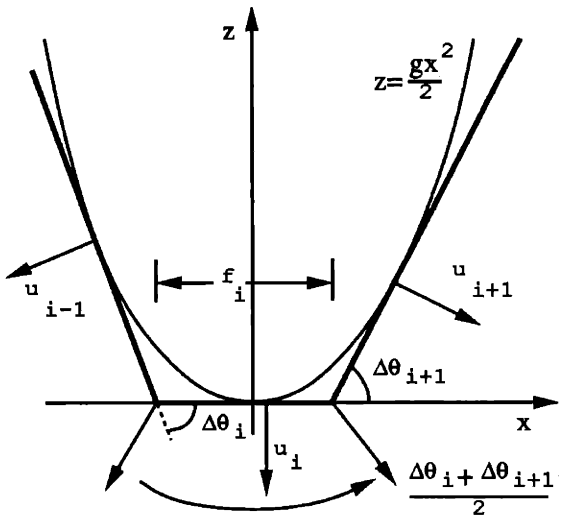

Figure 6-17: Definition 2 of discrete curvature, $\mathcal{K}_2$.

Definition $\mathcal{K}_2(i)$. The limiting problems seen above lead us to consider a different definition of discrete curvature. The previous definition was skewed to one side of the face. To correct this situation a face centered approach is considered, as illustrated in Figure 6-17. Here a derived "normal" is defined at each vertex which is the average of the adjacent face normals, $(u_i + u_{i-1})/2$ and $(u_{i+1} + u_i)/2$. The change in these *derived* vertex normals is then taken as the normal change associated with a face. This resulting angle change may be shown to be just the *average* of the adjacent angle changes $(\Delta\theta_i + \Delta\theta_{i+1})/2$. The resulting definition of curvature is then given by

$$\mathcal{K}_2(i) \equiv \frac{\Delta\theta_i + \Delta\theta_{i+1}}{2f_i}$$

The limiting curvature is now examined. Again, applying the definition of $\mathcal{K}_2$ to the quadratic boundary, we show in Appendix 6-D.2 that for this case one obtains:

$$\mathcal{K}_2(i) = \frac{g(\Delta\theta_i + \Delta\theta_{i+1})}{\tan(\Delta\theta_i) + \tan(\Delta\theta_{i+1})}$$

Again letting $\Delta\theta_{i+1} = \alpha\Delta\theta_i$, so the scalar α measures the relative rate of convergence of the two adjacent angles, we obtain in the limit as $\Delta\theta_i$ goes to zero:

$$\lim_{\Delta\theta_i \to 0} \mathcal{K}_2(i) = g.$$

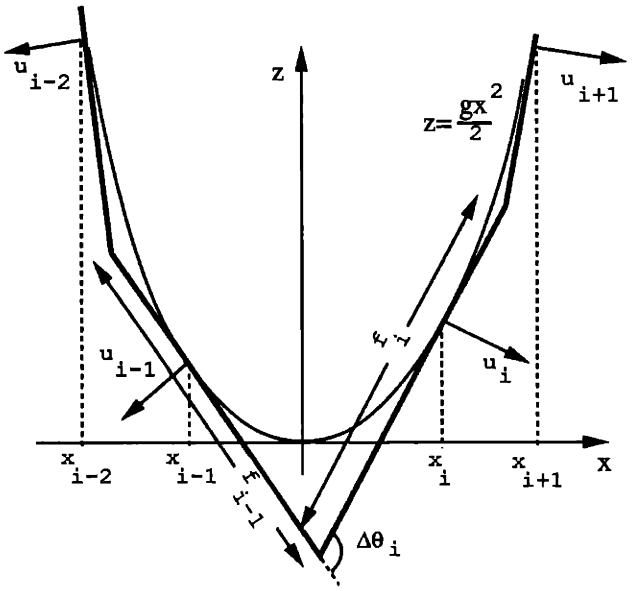

Figure 6-18: Definition 3 of discrete curvature, $\mathcal{K}_3$.

This expression is independent of α, the relative rate of convergence of the two angles. The definition $\mathcal{K}_2(i)$ should thus be more robust to the details of how our discrete, support based representation of an object approaches the underlying object. In general we would expect the curvature measure $\mathcal{K}_2(i)$ to be better behaved than the previous definition $\mathcal{K}_1(i)$. With this success, we try another definition, actually the dual of the present definition $\mathcal{K}_2(i)$.

Definition $\mathcal{K}_3(i)$. The dual formulation to that used for $\mathcal{K}_2(i)$ is now examined. Whereas the definition $\mathcal{K}_2(i)$ was a *face* centered definition, a *vertex* centered definition is now considered, as shown in Figure 6-18. Instead of averaging the turning angles and combining this average with the given face length as we did above, here the face lengths are averaged and combined with the angle change at a single vertex. The resulting definition of curvature is given by:

$$\mathcal{K}_3(i) \equiv \frac{2\Delta\theta_i}{f_{i-1} + f_i}$$

To analyze the limiting properties of this definition the definition $\mathcal{K}_3$ is again applied to a quadratic boundary. In Appendix 6-D.3 we show that for this case the discrete

curvature is:

$$\mathcal{K}_3(i) = \frac{4 \arctan\left(\frac{g(x_i - x_{i-1})}{g^2 x_i x_{i-1} + 1}\right)}{(x_{i+1} - x_i)\sqrt{(g^2 x_i^2 + 1)} + (x_i - x_{i-2})\sqrt{(g^2 x_{i-1}^2 + 1)}}$$

where the x_i are the x coordinates of the points of tangency, as shown in the figure. We work with the x_i for this case because of the difficulty of writing these expressions in terms of the $\Delta\theta_i$. Note that we now have a second order expression in the sample values, in the sense that support values up to two points away are involved. This involvement is necessary to establish the lengths of the segments f_i and f_{i-1}.

For convergence analysis let the different x coordinates be related as follows:

$$\begin{aligned} x_{i+1} &= x_i/\alpha \quad 0 \leq \alpha \leq 1 \\ x_{i-1} &= -\beta x_i \quad 0 \leq \beta \\ x_{i-2} &= x_{i-1}/\gamma \quad 0 \leq \gamma \leq 1. \end{aligned}$$

With these relationships, it is shown in Appendix 6-D.3 that as x_i goes to zero we obtain for the limiting curvature the value

$$\lim_{x_i \to 0} \mathcal{K}_3(i) = g \frac{4\alpha\gamma(\beta + 1)}{\alpha\beta\gamma + \alpha\beta + \alpha\gamma + \gamma}.$$

The proper curvature value of g is obtained if and only if

$$\frac{4\alpha\gamma(\beta + 1)}{\alpha\beta\gamma + \alpha\beta + \alpha\gamma + \gamma} = 1.$$

This expression defines a *family* of correct convergence routes, and it thus lies between the situation presented by the definitions of $\mathcal{K}_1(i)$ and $\mathcal{K}_2(i)$. In particular, one possible choice of the convergence parameters satisfying this constraint is $\beta = 1$ and $\alpha = \gamma = 1/3$. With this choice x_{i-1} and x_i approach the origin at equal rates from either side, and the outermost points are a factor of three farther away then the inner ones.

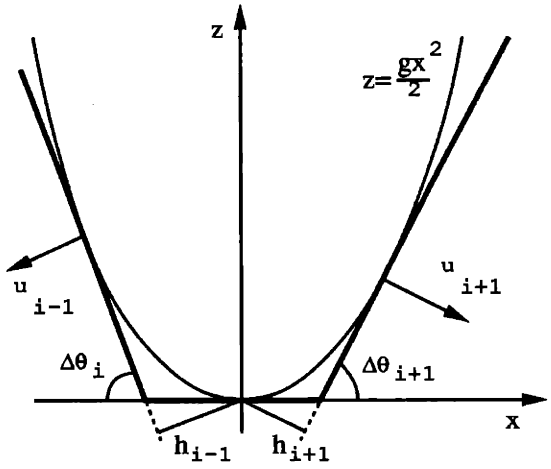

Figure 6-19: Definition 4 of discrete curvature, $\mathcal{K}_4$.

Definition $\mathcal{K}_4(i)$. The previous definitions of discrete curvature were primarily geometric in origin, based on direct formulations of the normal change per unit or arc length. Here an algebraic approach is examined. Recall that the curvature of a smooth planar object was given in terms of its support function in (6.2) by the expression

$$\mathcal{K}(\theta) = [h_{\theta\theta}(\theta) + h(\theta)]^{-1}$$

where $h_{\theta\theta}$ was the second derivative of the (reduced) support function with respect to θ. The present idea then is to use a second difference approximation to the term $h_{\theta\theta}$ to obtain a discrete analog to this formula. The resulting discrete curvature definition then becomes:

$$\mathcal{K}_4(i) \equiv \left[\frac{h_{k+1} - 2h_k + h_{k-1}}{\Delta\theta_{k+1}\Delta\theta_k} + h_k\right]^{-1}$$

We now apply this definition to the parabolic boundary to study its limiting behavior (see Figure 6-19). In Appendix 6-D.4 we show that the resulting curvature is given by the expression:

$$\mathcal{K}_4(i) = \frac{2g\Delta\theta_i\Delta\theta_{i+1}\cos(\Delta\theta_i)\cos(\Delta\theta_{i+1})}{\cos(\Delta\theta_i)\sin^2(\Delta\theta_{i+1}) + \cos(\Delta\theta_{i+1})\sin^2(\Delta\theta_i)}.$$

Setting $\Delta\theta_{i+1} = \alpha\Delta\theta_i$ and letting $\Delta\theta_i$ approach 0, we obtain for the limiting curvature

$$\lim_{\Delta\theta_i \to 0} \mathcal{K}_4(i) = g\frac{2\alpha}{\alpha^2 + 1}.$$

As for the definition $\mathcal{K}_1(i)$, the proper curvature is obtained in the limit if and only if $\alpha = 1$.

Our convergence analyses are meant to be illustrative of the potential problems with certain intuitive notions of discrete curvature. In particular for the equal angle case treated in [8] many different definitions of curvature yield correct limiting results, while in the general angle case one must be more careful. This is particularly troublesome for the 3-dimensional case. In three and higher dimensions the only regular tessellations of the sphere are the Platonic solids (or their generalizations [104]. The most finely tessellated of these, the icosahedron, has only 20 faces. In general then, for the higher dimensional cases it is impossible to achieve complete regularity in the limit, and so we desire curvature definitions that do not require it.

6.4.2 3-Dimensional Case

One approach we may take in the higher dimensional case is to couple the planar results of Section 6.4.1 together with the standard definition of planar normal curvature [71] to immediately develop discrete definitions of *surface* curvature. Such an approach yields an *extrinsic* definition of curvature. A discrete *intrinsic* notion of (Gaussian) curvature is also developed. Since explicit limiting analyses for the 3-dimensional cases appears much more complicated than for the planar case, we confine ourselves here to general remarks.

Planar Extensions: Extrinsic Curvature

Unlike the planar situation, in the 3-dimensional case there are multiple notions of surface curvature. One approach to capturing the curvature of a surface is to examine the properties of *curves* on the object's surface. In particular, the normal curvature of a surface at a point p is defined as the curvature of the planar curve obtained as the intersection of a plane containing the surface normal at p with the object, see Figure 6-20. This curvature is clearly a periodic function of the orientation of the intersecting plane about the normal. For a smooth convex object there are only two

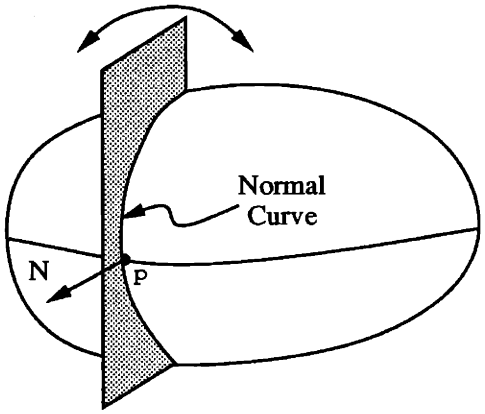

Figure 6-20: Illustration of normal curvature.

stationary points in the resulting curvature value as the plane is turned around the normal. These values correspond to special orientations of the plane which are at right angles to each other. The resulting maximum and minimum values of curvature are termed the *principal curvatures* of the surface and the corresponding directions of the intersecting plane the *principal directions* [71]. The principal radii of curvature R_k were discussed earlier in connection with the support function. These values are just the reciprocals of the principal curvatures.

The principal curvatures and directions so defined appear in different definitions of *surface* curvature. The two principal values themselves are often used as a measure of how much the surface curves in certain directions. In fact, we may obtain the plane curvature for *any* orientation of the intersecting plane in Figure 6-20 from knowledge of only the principle directions and curvatures (a result known as Euler Theorem [73]). These values thus completely characterize the curvature of the surface at p. Further, knowledge of these values is equivalent to knowledge of the Hessian of the surface at p and, as we discussed in Chapter 4, completely characterizes a convex surface if it is known at all points (at least to within translation). Often scalar functions of the principal values are used to obtain a scalar measure of the surface curvature, with the associated directional information discarded. A common function is the average of the principal curvatures, termed the *mean curvature*. These definitions of surface curvature are extrinsic, in that they depend on the embedding of the object in the underlying space. For example, these definitions give different values for the curvature of a cylinder and a plane, surfaces that cannot be distinguished from local measurements alone.

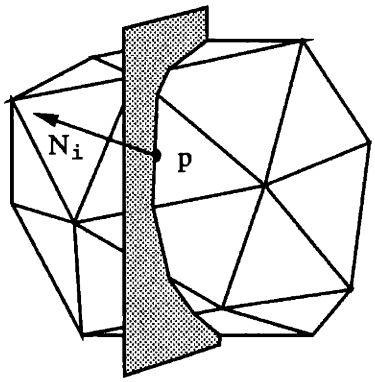

Figure 6-21: Illustration of discrete planar curvature.

The above notions of surface curvature directly depend on planar definitions of curvature through the use of the intersecting plane. In the same way, any of the planar definitions of discrete curvature may be immediately coupled with an intersecting plane to develop a corresponding discrete definition of surface curvature. The intersection of such a plane, containing the surface normal of a face, with the polyhedral object resulting from a set of (consistent) support measurements will result in a plane polygonal curve, as illustrated in Figure 6-21. A planar definition of discrete curvature applied to this polygonal curve yields a discrete normal curvature. The overall maximum and minimum of the values of this curvature obtained as the plane rotates around the surface normal may then be taken as definitions of the *discrete principal curvatures.* In this operation we take the surface normal to be at the center of each face. An interesting question is the relationship between these (perhaps multiple) maxima and minima and the underlying polyhedron. For example, a regular cube will have two maxima, corresponding to orientations of the plane perpendicular to each edge, and a two minima, corresponding to orientations of the plane along each diagonal.

This definition of discrete surface curvature is well defined if the corresponding definition of discrete planar curvature is. The discrete plane curvatures will converge to their true values as the polyhedral approximation approaches the underlying shape if and only if the planar definitions also have this property. In particular, functions of the principle curvatures, such as the mean curvature, also approach their correct values as the discrete planar curvatures approach their individual correct values. As a result, our analysis of the planar case is of direct use for these definitions of *surface* curvatures.

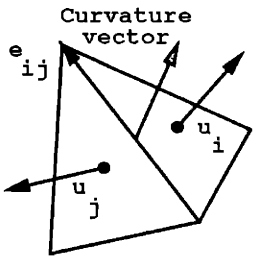

Figure 6-22: Mean curvature vector.

We mention another discrete definition of the extrinsic mean curvature which does not depend on the above planar arguments. Suppose e_{ij} is the directed edge between faces of a polyhedron with unit normals u_i and u_j, where the orientation of the edge is induced by the normals, as shown in Figure 6-22. In [105] the mean curvature vector is then defined to be

$$\frac{1}{2} e_{ij} \times (u_i - u_j)$$

where $\times$ denotes cross product. Now e_{ij} and $(u_i - u_j)$ are perpendicular so the magnitude of this vector is the edge length times half the length of the vector between the adjacent face normals (which is at most 2 since the normals are unit vectors). Since the vertex positions are linear functions of the support vector $\mathbf{h}$, so is the edge vector e_{ij}. As a result the edge lengths (squared) are a quadratic function of the support vector $\mathbf{h}$. Finally, note that this definition of curvature does not reduce to any of the planar definitions when applied to polygons.

Direct Methods: Intrinsic Curvature

Direct definitions of curvature for the 3-dimensional case are now examined. These definitions are extensions of, but do not use, the planar concepts of Section 6.4.1. In the previous section, a natural discrete version of the extrinsic normal curvature of a surface was developed. Such a definition may also be used to define discrete versions of such functions of the principal curvatures as the mean curvature. Here we aim to extend the *intrinsic* concept of the Gaussian curvature to the discrete case. Geometrically, the Gaussian curvature $\mathcal{K}$ is the limiting ratio of an area patch $d\omega$ on the Gaussian sphere (the unit sphere) and the corresponding infinitesimal area element df on the object, as shown in Figure 6-23. Thus the Gaussian curvature at a surface point with normal N is given by $\mathcal{K}(N) = d\omega/df$. Note that, as mentioned previously, the Gaussian curvature may also be obtained algebraically as the *product* of the principle curvatures, yielding another route to its definition than the one taken

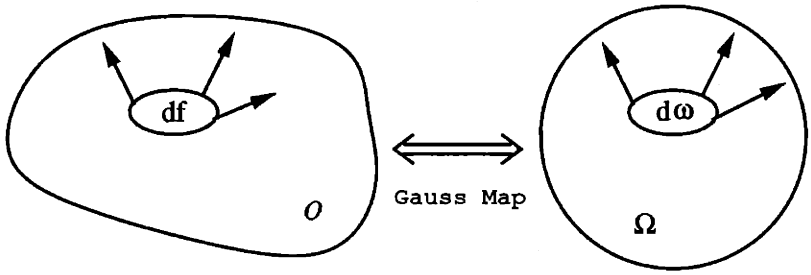

Figure 6-23: Illustration of Gaussian curvature.

here.

The Gaussian curvature is an *intrinsic* property of the surface, measuring the deviation of the surface locally from flatness. This means that only surface points that can be disguised by local measurements will have different Gaussian curvatures, in contrast to the case for the mean curvature (an extrinsic measure). For example, a plane and a cylinder both have zero Gaussian curvature, since locally they appear the same, yet their mean curvature at any point is different. In other words, the value of the Gaussian curvature does not depend on the embedding of the surface. Note that $\mathcal{K}$ is a scalar function, and hence a non-directional measure.

In contrast to the concepts of curvature discussed in the previous section then, the Gaussian curvature may be defined directly from geometrical considerations, without resorting to planar concepts. A discrete version of the prescription given in Figure 6-23 and by the formula $\mathcal{K}(N) = d\omega/df$ is used. As in the planar case, we may center our definition at either a face or a vertex. Recall that in the planar case, the face centered definition seemed to be better behaved and so the equivalent face centered definition is presented first.

Intuitively, all the curvature in the discrete, polyhedral object representations under consideration is contained in the object's vertices. Each such vertex corresponds to a solid angle on the Gaussian sphere and conversely, all the area of Gaussian sphere is mapped to the vertices. In contrast, on the object, all the area is restricted to the faces, which correspond to mere points on the Gaussian sphere. For a discrete approximation we must devise a way to spread the curvature at the vertices over the area of the surface in a reasonable way, so that in the end the total curvature is distributed over the total surface area. This is the same approach as taken for our planar definitions.

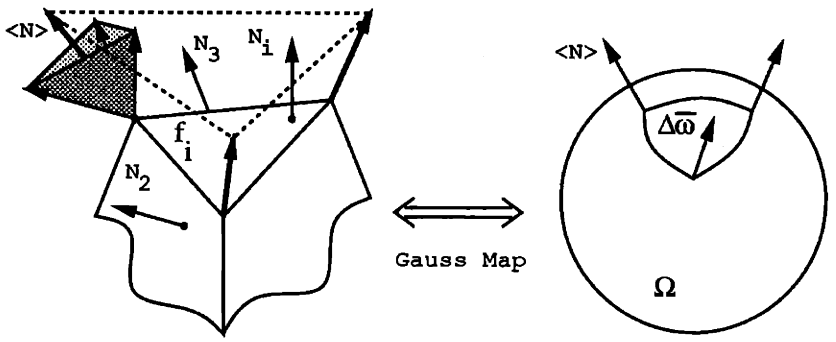

Figure 6-24: Illustration of definition $\mathcal{K}_5(i)$.

Definition $\mathcal{K}_5(i)$. For the face centered definition, the area f_i of each face is left alone and the cone of normals at each vertex is apportioned to the adjacent faces, see Figure 6-24. This approach is a generalization of the idea behind the planar definition $\mathcal{K}_2(i)$, and reduces to that definition for a planar object. We proceed by assigning to each vertex a normal which is the average of all the face normals adjacent to this vertex. The curvature area associated with face i, $\Delta\overline{\omega}_i$, is then defined to be the area of the spherical polygon (solid angle) on the Gaussian sphere corresponding to these derived vertex normals, as shown. The discrete (Gaussian) curvature is then defined as

$$\mathcal{K}_5(i) \equiv \frac{\Delta\overline{\omega}_i}{f_i}.$$

Geometrically, this operation is equivalent to evenly dividing the solid angle of curvature at a vertex equally between the adjacent faces.

Definition $\mathcal{K}_6(i)$. For completeness a vertex centered definition, dual to the above, is presented here. In contrast to the above approach, the cone of normals at a vertex is now left alone and the area of the adjacent faces is divided up, see Figure 6-25. Such an approach is a generalization of the idea behind the planar vertex centered definition $\mathcal{K}_3(i)$, and again reduces to this definition for planar objects. To be precise, the curvature area associated to a vertex, $\Delta\omega_i$ is now defined to be the area of the spherical polygon (solid angle) on the Gaussian sphere corresponding to the cone of normals at the vertex. To obtain the corresponding object surface area, the area of each face is now apportioned to its adjacent vertices by a uniform central tessellation. This tessellation is achieved by connecting the area centroid of a face with the center of each of its sides and assigning to a given vertex those segments touching it, as shown. The overall object surface area associated with the i-th vertex normal cone

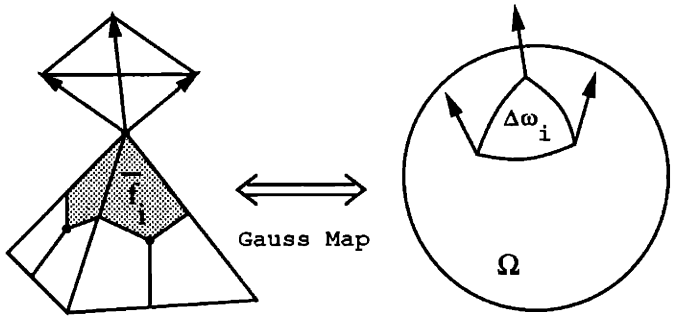

Figure 6-25: Illustration of curvature definition $\mathcal{K}_6(i)$.

$\overline{f}_i$ is then the sum of these surface area pieces from each face. The resulting discrete (Gaussian) curvature is then defined as

$$\mathcal{K}_6(i) \equiv \frac{\Delta\omega_i}{\overline{f}_i}.$$

Geometrically, this curvature definition is equivalent to evenly dividing the area of the faces among the adjacent vertices. Since any face may be triangulated by joining pairs of its vertices, we may assume that all faces are triangular without loss of generality. The facial tessellation required to find $\overline{f}_i$ then splits each face into three parts of equal area, each adjacent to a vertex. As a result, the derived object surface area $\overline{f}_i$ is just 1/3 of the sum of the areas of all the faces adjacent to vertex i:

$$\overline{f}_i = \frac{1}{3} \sum_{\substack{\text{Adjacent} \\ \text{to vertex } i}} f_j$$

where f_j is the area of face j. This calculation is a simple averaging operation dual to the averaging of the normals in the definition of $\mathcal{K}_5(i)$.

An important descriptor associated with each vertex is the *spherical deficit*, which is the difference between the circumference of a unit circle in the plane (2π) and the circumference of a unit circle on the object. For a polyhedron this quantity is given at a vertex by

$$\text{Spherical Deficit} = 2\pi - \sum_{\substack{\text{Adjacent} \\ \text{to vertex } i}} \psi_j$$

where $\sum \psi_j$ is the sum of the face angles at the vertex, as shown in Figure 6-26 [106,

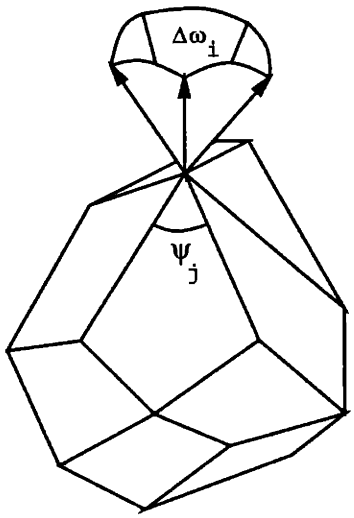

Figure 6-26: Relation between spherical deficit and normal cone.

pg. 128]. Since the solid angle of the normals at a vertex $\Delta\omega_i$ is equal to the spherical deficit at the vertex [106], the curvature $\mathcal{K}_6(i)$ is just the ratio of the spherical deficit at a vertex to 1/3 of the total area of the surrounding faces $\overline{f}_i$:

$$\mathcal{K}_6(i) = \frac{2\pi - \sum_{\substack{\text{Adjacent} \\ \text{to vertex } i}} \psi_j}{\frac{1}{3}\sum_{\substack{\text{Adjacent} \\ \text{to vertex } i}} f_j}. \tag{6.13}$$

The formula (6.13) also appears in the book *Difference Geometry* (in German) [107, pg. 13]. Unfortunately, [107], which develops differential geometry from the discrete difference standpoint, is not available in translation and further, [73, pg. 231] warns the reader against numerous typos in the work! The polyhedral object treated in [107] is obtained from an underlying surface by a uniform triangulation in parameter space. If the radius vector to a surface point p is given as a function of two parameters r, s by $p(r, s)$, then samples are taken at the points $r = r_0 \pm i\epsilon$ and $s = s_0 \pm j\epsilon$ for all integers i, j, where $p(r_0, s_0)$ is assumed the point of interest. The resulting surface samples form the vertices of the polyhedron. In [107] the claim appears to be made that as $\epsilon \to 0$ the discrete curvature measure defined by $\mathcal{K}_6$ approaches the Gaussian curvature at $p(r_0, s_0)$. From comments in [73] it also appears that the formula (6.13) can also be phrased in terms of the dihedral angles between the faces.

To understand the similarities and differences of the situation in [107] with ours we

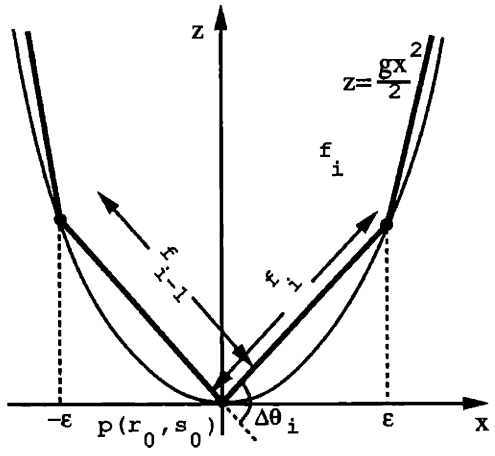

Figure 6-27: Polygonal approximation of [107].

consider the planar case. In Figure 6-27 is shown the triangulation of [107] applied to the quadratic surface we had used for studying planar limiting arguments. Applying the prescription of $\mathcal{K}_6$ to this planar case results in a discrete curvature at $p(r_0, s_0)$ given by $\mathcal{K}_3 = 2\Delta\theta_i/(f_i + f_{i-1})$. It is straightforward to show that as $\epsilon \to 0$ the discrete curvature $2\Delta\theta_i/(f_i + f_{i-1}) \to g$, the proper value. The corresponding vertex based definition, $\mathcal{K}_3$ for support samples, shown in Figure 6-18, does *not* approach the correct limiting value except for certain special cases.

These differences seem to result from the different forms of the underlying measurements. Since in [107] the vertices lie on the object surface, these points form natural focal points for discrete definitions. In contrast, for our support based case it is the *faces* that touch the surface (at unknown points) and seem to be the natural focus. The two types of sampling seem to be duals of one another, with faces replacing vertices. For convex objects the surface triangulation yields a polyhedral set which is contained *within* the object while the support based data yields one which contains the object. It seems reasonable that the properties of the various approximations are also dual. For example, as we saw above the vertex centered definition of curvature appears to behave well for a surface triangulation while the face centered definition appears better for the support based polyhedron. We leave further development of such ideas for future work.

Consistency

The Gauss-Bonnet theorem of differential geometry states that for any closed compact object surface $\partial\mathcal{O}$ the curvature function must satisfy the relation [71]:

$$\int_{\partial\mathcal{O}} \mathcal{K}\, df = 2\pi\chi \tag{6.14}$$

where df is the area element of the surface and χ is the Euler-Poincaré characteristic of the object. The Euler-Poincaré characteristic of all connected compact surfaces homeomorphic to a sphere (as are all those under consideration here) is $\chi = 2$. Thus a valid curvature function of a surface must satisfy this constraint.

Now for any polyhedron it can be shown that [106]:

$$\sum_{\substack{\text{Entire}\\ \text{Polyhedron}}} \psi_j = 2\pi\chi.$$

From this expression it is easy to see that the discrete version of (6.14) is always satisfied by $\mathcal{K}_6(i)$ since

$$\sum_i \mathcal{K}_6(i)\overline{f}_i = \sum_{\substack{\text{Entire}\\ \text{Polyhedron}}} \psi_j = 2\pi\chi.$$

Thus $\mathcal{K}_6(i)$ has the nice property of consistency with the constraint (6.14) of the continuous case (for an appropriately defined area element).

From the definition of $\mathcal{K}_5(i)$ and the fact that the solid angle of the normals at a vertex $\Delta\omega_i$ is equal to the spherical deficit at the vertex, it is straightforward to show that

$$\sum_i \mathcal{K}_5(i) f_i = \sum_{\substack{\text{Entire}\\ \text{Polyhedron}}} \Delta\omega_i = \sum_{\substack{\text{Entire}\\ \text{Polyhedron}}} \psi_j = 2\pi\chi$$

and so $\mathcal{K}_5(i)$ also is consistent (6.14) (again, for an appropriately defined area element). Note that in demonstrating the consistency of both $\mathcal{K}_5(i)$ and $\mathcal{K}_6(i)$ we chose the corresponding area element conveniently, in one case choosing f_i and in the other $\overline{f}_i$.

It is difficult to say whether the planar based surface curvature definitions have

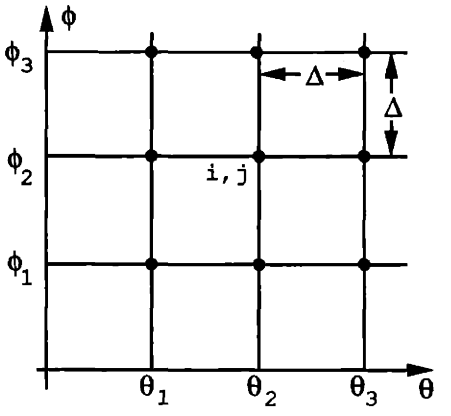

Figure 6-28: Square grid of sample points.

this property in general. If the planar discrete curvature functions converge to the underlying surface curvature in the limit, then certainly the surface curvature measures will also be consistent in the limit.

Algebraic Approaches

Some algebraic approaches to obtaining discrete definitions of surface curvature are examined here. These definitions arise primarily from algebraic concerns, in contrast to the geometrically based definitions above. These methods appear less promising than the above, but are included for completeness.

Finite Differences One approach is to discretize the corresponding continuous formula directly using finite difference approximations, as was done for $\mathcal{K}_4$. Recall that the continuous expression for the curvature was given in terms of the spherical angular coordinates θ, ϕ, and the (reduced) support function $h(\theta, \phi)$ by:

$$1/\mathcal{K}(\theta,\phi) = (h + h_{\theta\theta})\left(h_{\phi\phi}\csc^2(\theta) + h_\theta \cot(\theta) + h\right) - \csc^2(\theta)\left(h_{\theta\phi} - h_\phi \cot(\theta)\right)^2$$

Suppose, for simplicity, that the support samples are taken on a square grid, uniform in both θ and ϕ, as shown in Figure 6-28. Let the regular spacing between the neighboring θ and ϕ points be given by Δ and $h_{i,j} = h(\theta_i, \phi_j)$. At grid point i, j we may now make the following substitutions for the continuous elements of the equation

to define a discrete curvature value at the point [108]:

$$\begin{aligned}
h &\sim h_{i,j} \\
h_\theta &\sim (h_{i+1,j} - h_{i-1,j})/2\Delta \\
h_\phi &\sim (h_{i,j+1} - h_{i,j-1})/2\Delta \\
h_{\theta\theta} &\sim (h_{i+1,j} - 2h_{i,j} + h_{i-1,j})/\Delta^2 \\
h_{\phi\phi} &\sim (h_{i,j+1} - 2h_{i,j} + h_{i,j-1})/\Delta^2 \\
h_{\theta\phi} &\sim (h_{i+1,j+1} - h_{i+1,j-1} + h_{i-1,j-1} - h_{i-1,j+1})/(2\Delta)^2
\end{aligned}$$

The trigonometric functions are assumed to be evaluated at $\theta = \theta_i$. The resulting discrete equation is not linear in the samples $h_{i,j}$ due to terms involving the product of samples. These products arise in the 3-dimensional case because of the interaction of the spherical coordinates θ and ϕ, a difficulty with any parameterization of the sphere which does not exist for the planar case. As a consequence, the formula obtained in the above procedure does not seem to simplify much and is certainly less intuitive geometrically than the previous discrete definitions.

Other finite difference transformations of continuous equations may also be done, including extensions to irregular sampling grids, a more realistic assumption in practice. A wealth of literature exists on the general area of finite difference approximations and solutions of continuous partial differential equations, and a deeper treatment of the subject is beyond the scope of this work. The reader is referred to e.g. [108, 109] for more detail on these approaches.

Fitting Quadrics Another approach to extracting curvature information from a discrete set of measurements is discussed in [73, pg. 321]. An osculating (best fitting) quadric of the form $z = [x\ y]\mathsf{H}[x\ y]^T$ is fit to the set of measurements in local coordinates, as shown in Figure 6-29. The symmetric matrix H defining this quadric in local tangential coordinates is the Hessian or the second second fundamental form of the surface [71, 73]. As we discussed in Chapter 4 the Gaussian curvature is given by its determinant and the principal curvatures and directions by its eigenvalues and vectors. Thus finding this matrix effectively gives curvature information about the surface.

In the present setting the samples are provided by normal/support value pairs (h_i, u_i), where u_i is the unit normal associated with sample h_i. Given such samples

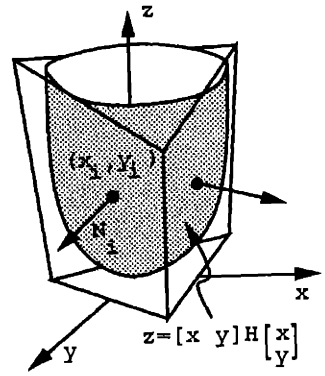

Figure 6-29: Quadratic fit to samples.

of the above quadric, the follow relationship holds:

$$\frac{h_i}{[\ 0\ \ 0\ \ -1\ ]u_i} = [\ x_i\ \ y_i\ ]\mathsf{H}\begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{6.15}$$

where (x_i, y_i) are the x, y coordinates of the point of tangency of the support plane. Since these points of tangency (which are not given by support samples) *and* the parameters of the quadratic must be estimated, the resulting problem is nonlinear, involving the *product* of linear terms in the curvature parameters and quadratic terms in the tangency points. The approach thus seems ill suited to the present setting, and our focus instead has been to work directly with the support measurements in developing definitions of boundary curvature. Such potential difficulties were noted in [73], but for the opposite situation where the tangency points are known but the normals are unknown, resulting in a bilinear problem.

If both the normal samples u_i *and* the tangency points (x_i, y_i) are somehow known, then finding H above reduces to the linear problem treated in Chapter 3, and all the techniques developed there may be used. This situation of known tangency points is the one treated in [73], where both surface samples (x_i, y_i, z_i) and normals are obtained from direct sampling of the surface through a triangulation. It appears from comments in [73] that the linear equations obtained from a surface triangulation were solved symbolically in the German book [107], discussed in connection with the definition of $\mathcal{K}_6$.

6.4.3 Use in Optimization

The curvature definitions considered in this section have all represented *local* properties of the surface. As a measure of smoothness, it is the maximum value of such a curvature function over the surface that is important. To find the smoothest estimated support vector $\hat{\mathbf{h}}$ for a given set of observations we might imagine finding the consistent support vector whose maximum curvature value is minimized. This formulation has several difficulties, however. The solution to the problem as stated is unbounded, since for a given shape a uniform increase in object size reduces the curvature at all points. The difficulty is that curvature directly depends on object scale. This situation may be partially remedied through the requirement that the estimate $\hat{\mathbf{h}}$ come from a predefined bounded set $\mathcal{B}$. Such an approach was taken in [8], through the imposition of a bounded noise model. Even with this addition, note that the solution does not depend on our observations $\mathbf{h}_{\text{obs}}$.

To include the data in the problem we may use a cost function which combines both the distance of our estimate $\hat{\mathbf{h}}$ from the observations together with the maximum curvature of the estimate, $\max \mathcal{K}(i)$. For example, we could find a support vector estimate as the solution to:

$$\hat{\mathbf{h}} = \underset{\bar{Q}\mathbf{h}\geq 0,\, \mathbf{h}\in\mathcal{B}}{\arg\min} \left[(1-\alpha)\|\mathbf{h}-\mathbf{h}_{\text{obs}}\|_2^2 + \alpha \max_i \mathcal{K}(i)\right]$$

where the constraint $\bar{Q}\mathbf{h} \geq 0$ imposes consistency on the estimate, $\mathcal{K}(i)$ is the i-th discrete curvature value from any of the discrete definitions above, and $0 \leq \alpha \leq 1$ is a parameter which allows a tradeoff between smoothness and adherence to data. Note that $\mathcal{K}(i)$ is a function of the support vector $\mathbf{h}$. When $\alpha = 0$ the closest consistent support vector in the set $\mathcal{B}$ is produced as the estimate, with no concern for smoothness of the estimate. At the other extreme, as $\alpha \rightarrow 1$ the data matters less and less, and the curvature of the estimate becomes the primary concern.

This estimate still has the undesirable property that it depends on the prior choice of the bounding set $\mathcal{B}$. One solution to this difficulty is to remove the set constraint $\mathcal{B}$ and then perform the optimization over support vectors of a fixed size, for example by requiring that $\hat{\mathbf{h}}$ have unit norm. The optimal size of the vector could then be estimated jointly or in a separate step. This approach was taken for the planar case in [8], where effectively the 1-norm of the support vector was held constant. This quantity is proportional to the circumference of the object in the planar case. In the

next section we examine notions of smoothness which do not depend on the local properties of the object.

6.5 Global Smoothness Concepts

In contrast to the above curvature-based paradigm, a *global* measure of smoothness is developed in this section. The classic isoperimetric inequality is used to define a discrete measure of overall object smoothness. This measure indicates the deviation of an object from a sphere, which is in some sense the smoothest object of a given volume. The measure has the desirable property that it is independent of object scale.

Notions of smoothness based on curvature properties are inherently local, depending on point properties of the surface. Obtaining such curvature information at all points simply to impose a smoothness condition is potentially inefficient. Often, we do not care about the details of the distribution of this curvature. A reconstruction using an overall measure of smoothness would then be sufficient. Such a measure is provided by an isoperimetric ratio, which gives a measure of the deviation of an object from a sphere (or circle), the object of minimal surface area for a given volume, and hence the smoothest in this sense.

6.5.1 The Planar Isoperimetric Inequality

It is well known that in the plane, the circle has the smallest circumference of all regions of given area. This statement is reflected in the classic isoperimetric inequality

$$(\text{Perimeter})^2 \geq 4\pi(\text{Area}) \tag{6.16}$$

where equality holds only in the case of a circle [110]. In the plane, then, we may think of trying to minimize a measure of roughness of a surface as provided by:

$$\mathcal{R} = \frac{(\text{Perimeter})^2}{(\text{Area})}.$$

which we term the isoperimetric ratio. Such a quantity measures the distance of an object from a circle and achieves its minimum only for a circle. In contrast to the

curvature, $\mathcal{R}$ is independent of scale. There is no tendency to enlarge an object in the above minimization process as there is in attempting to minimize the maximum curvature point.

As an alternative to $\mathcal{R}$, we could also use the quantity

$$\widetilde{\mathcal{R}} = (\text{Perimeter})^2 - 4\pi(\text{Area}) \geq 0$$

termed the *isoperimetric deficit*, as a measure of roughness. This quantity is used in [111] in the study of stability of geometric inequalities. The same ideas apply to either of these definitions. In some cases one might be preferable over the other for reasons of computational tractability. In particular, the application of these measures to discrete polyhedral objects are different in their sensitivity to scaling.

6.5.2 The General Case

A result similar to (6.16) exists in higher dimensions. For a convex body in R^n the following inequality holds:

$$\mathcal{R} = \frac{f^n}{v^{n-1}} \geq \frac{\omega_n^{n-1}}{\kappa_n^n} = n^n \kappa_n$$

where f is the surface area ($(n-1)$-dimensional volume) of the surface of the object, v is its volume, and ω_n and κ_n are the area and volume of the n-dimensional unit ball, respectively. The last equality follows from the fact that $\omega_n = n\kappa_n$. This generalization is originally due to Minkowski, and may be shown to be a consequence of the Brunn-Minkowski theorem (see e.g. [27]). The inequality only holds for *convex bodies* in the higher dimensional case, apparently because, unlike the planar case, in higher dimensions the convex hull of an arbitrary body can actually have a larger value of surface area than the original object [27]. The equivalent generalization of $\widetilde{\mathcal{R}}$ is naturally given by

$$\widetilde{\mathcal{R}} = f^n - n^n \kappa_n v^{n-1} \geq 0.$$

6.5.3 Polyhedra

The above isoperimetric inequalities are in terms of the volume and surface area of the object of interest. Since the objects under consideration, arising from discrete support samples, are polyhedral, simple formulas for the area and volume of polyhedra are needed to proceed. Versions of the continuous formulas presented in Section 6.2.3 are given here for the discrete case of convex polygons and polyhedra. Let $f_i(u_i)$ be the area ($(n-1)$-dimensional volume) of the face with unit normal u_i. In the plane the f_i are the lengths of the sides of the polygon, while for the 3-dimensional case, the f_i are the areas of the faces of a polyhedron. As usual, let $H(u_i) = h_i$ be the support value of the polytope in direction u_i, the distance of the i-th face from the origin. We assume there are m faces.

With this notation, we have for the surface area of a polyhedron:

$$\text{Surface Area } = f = \sum_{i=1}^{m} f_i$$

The n-dimensional volume of the polytope is given by the discrete version of (6.3) [27, 92]:

$$\text{Volume } = v = \frac{1}{n} \sum_{i=1}^{m} h_i f_i$$

In the planar case the surface area becomes the perimeter of the polygon and the n-dimensional volume becomes the enclosed area. For the 3-dimensional case the quantities are indeed the surface area and volume respectively.

Substituting these values into the smoothness formulas above, the general isoperimetric ratio $\mathcal{R} = f^n / v^{n-1}$ then becomes for polyhedra:

$$\mathcal{R} = \frac{\left(\sum_{i=1}^{m} f_i\right)^n}{\left(\frac{1}{n} \sum_{i=1}^{m} h_i f_i\right)^{n-1}}$$

and the corresponding expression for the isoperimetric deficit is given by:

$$\widetilde{\mathcal{R}} = \left(\sum_{i=1}^{m} f_i\right)^n - n\kappa_n \left(\sum_{i=1}^{m} h_i f_i\right)^{n-1}$$

where, again, n is the dimension of the space, h_i is the support value associated with

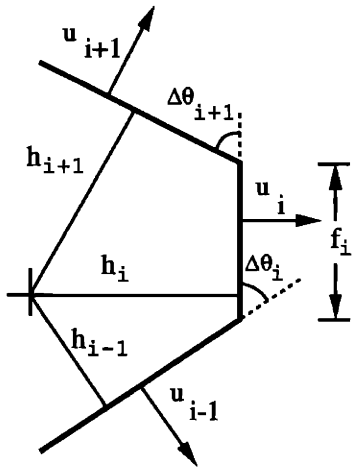

Figure 6-30: Calculation of f_i.

face i, f_i is the area (or length) of face i, and κ_n is the volume (or area) of the n-dimensional unit ball.

As it stands, the expression $\widetilde{\mathcal{R}}$ applied to a polyhedra is minimized for a zero size object (one with $\mathbf{h} = 0$), reflecting a size dependence. In particular for a fixed object shape, $\widetilde{\mathcal{R}}$ grows as $\|\mathbf{h}\|_2^2$. In the continuous case the lower bound of zero was achieved for *any* sphere, regardless of size. The problem is that for convex polyhedra of given (nonzero) volume, the sphere of the same volume always has a strictly lower deficit, namely zero. Thus for polyhedra the deficit $\widetilde{\mathcal{R}}$ is positive for polyhedra of positive volume and achieves zero only when the polyhedral volume is also. A simple solution to the scale problem for the polyhedral case is to normalize the quantity $\widetilde{\mathcal{R}}$ by the size of the support vector, given by its squared norm $\sum_{i=1}^m h_i^2$:

$$\widehat{\mathcal{R}} = \frac{\left(\sum_{i=1}^m f_i\right)^n - n\kappa_n \left(\sum_{i=1}^m h_i f_i\right)^{n-1}}{\sum_{i=1}^m h_i^2}.$$

The 2-Dimensional Case

Here we apply the formulas for $\mathcal{R}$ and $\widehat{\mathcal{R}}$ to the planar, arbitrary angle case to illustrate their use. The inverse of the measure $\mathcal{R}$ (but without this interpretation) was used for the planar equal angle case in [8] to eliminate the growth tendency of curvature minimization.

For the arbitrary angle case it is straightforward to show that the i-th face length f_i is given by the linear function (see Figure 6-30):

$$f_i = \begin{bmatrix} \frac{1}{\sin(\Delta\theta_i)} & -\frac{\sin(\Delta\theta_i+\Delta\theta_{i+1})}{\sin(\Delta\theta_i)\sin(\Delta\theta_{i+1})} & \frac{1}{\sin(\Delta\theta_{i+1})} \end{bmatrix} \begin{bmatrix} h_{i-1} \\ h_i \\ h_{i+1} \end{bmatrix}$$

where, recall, $\Delta\theta_i = \theta_i - \theta_{i-1}$. Note that this expression is very much like a row of $\bar{Q}$ in the support consistency constraint (6.11), differing only in positive scaling. Thus we may interpret the consistency constraint $\bar{Q}\mathbf{h} \geq 0$ as a statement that the facial lengths must be non-negative $f_i \geq 0$.

We may combine the above individual relationships for the facial lengths to write:

$$\mathbf{f} = F\mathbf{h} \tag{6.17}$$

where $\mathbf{f} = [f_1, f_2, \cdots, f_m]^T$ is the vector of face lengths, $\mathbf{h}$ is the support vector, and F is the symmetric matrix

$$F = \begin{bmatrix} -\frac{\sin(\Delta\theta_1+\Delta\theta_2)}{\sin(\Delta\theta_1)\sin(\Delta\theta_2)} & \frac{1}{\sin(\Delta\theta_2)} & 0 & \cdots & \frac{1}{\sin(\Delta\theta_1)} \\ \frac{1}{\sin(\Delta\theta_2)} & -\frac{\sin(\Delta\theta_2+\Delta\theta_3)}{\sin(\Delta\theta_2)\sin(\Delta\theta_3)} & \frac{1}{\sin(\Delta\theta_3)} & & 0 \\ 0 & \frac{1}{\sin(\Delta\theta_3)} & -\frac{\sin(\Delta\theta_3+\Delta\theta_4)}{\sin(\Delta\theta_3)\sin(\Delta\theta_4)} & & \vdots \\ \vdots & 0 & \frac{1}{\sin(\Delta\theta_4)} & & 0 \\ 0 & \vdots & & \cdots & \frac{1}{\sin(\Delta\theta_m)} \\ \frac{1}{\sin(\Delta\theta_1)} & 0 & \cdots & & -\frac{\sin(\Delta\theta_1+\Delta\theta_m)}{\sin(\Delta\theta_1)\sin(\Delta\theta_m)} \end{bmatrix}.$$

In the case of equal $\Delta\theta_i$, the matrix is circulant. Substituting the relationship (6.17) into the expression for the isoperimetric ratio $\mathcal{R}$ yields:

$$\mathcal{R} = \frac{2\mathbf{h}^T F W F \mathbf{h}}{\mathbf{h}^T F \mathbf{h}}$$

where W is an $n \times n$ matrix of ones. In the case that the angle increment is uniform this expression may be simplified slightly.

The equivalent expression for the normalized isoperimetric deficit $\widehat{\mathcal{R}}$ is given by

$$\widehat{\mathcal{R}} = \frac{\mathbf{h}^T (FWF - 2\pi F)\mathbf{h}}{\mathbf{h}^T \mathbf{h}}.$$

Thus $\widehat{\mathcal{R}}$ is just a quadratic form in the support vector $\mathbf{h}$.

Minimization

Minimization of $\mathcal{R}$ is a variational problem with interesting and natural geometrical interpretations. The minimum of $\mathcal{R}$ is found from the solutions of the following generalized eigenvalue problem:

$$2FWFu = \lambda Fu. \tag{6.18}$$

If λ_i, u_i are the eigenvalue/eignvector pairs of this problem, then the minimum value of $\mathcal{R}$ is given by the minimum positive eigenvalue (since we know the quantity must be positive for a polygon of finite area) and this minimum is achieved for the corresponding eigenvector. The problem (6.18) has only one nonzero eigenvalue, which must correspond to the minimizing value. Let e be the vector of n ones, so that $W = ee^T$. It is straightforward to show that $u = e$ is the eigenvector of (6.18) corresponding to the positive eigenvalue $1/2e^TFe$, so that e must be a minimizing support vector.

Now it is straightforward to show that the matrix F is singular and that the following two vectors are in its nullspace:

$$\begin{bmatrix} \cos(\theta_1) & \cos(\theta_2) & \cdots & \cos(\theta_m) \end{bmatrix}$$
$$\begin{bmatrix} \sin(\theta_1) & \sin(\theta_2) & \cdots & \sin(\theta_m) \end{bmatrix}$$

Vectors in the nullspace correspond to translations of the object [8]. Now adding any vector h_{Null} from this nullspace to the vector e does not change the value of $\mathcal{R}$. This makes geometrical sense, since translating an object does not change its area or perimeter. For a given set of sample directions then, the support vectors $\alpha e + h_{\text{Null}}$, for any positive scalar α, thus corresponds to the smoothest objects in the sense under consideration. Geometrically these are just support vectors from shifted circles.

The minimization of $\widehat{\mathcal{R}}$ yields similar results. The expression for $\widehat{\mathcal{R}}$ is a Rayleigh quotient and the minimum is found from the solutions of the corresponding eigenvalue problem:

$$(FWF - 2\pi F)u = \lambda u. \tag{6.19}$$

If λ_i, u_i are the eigenvalue/eigenvector pairs of this problem, then, as above, the minimum value of $\widehat{\mathcal{R}}$ is given by the minimum positive eigenvalue of (6.19) (since

again we know the quantity must be positive) and this minimum is achieved for the corresponding eigenvector. The problem (6.19) now has more than one nonzero eigenvalue, but again it is straightforward to show that $u = e$, the vector of ones, is an eigenvector with corresponding eigenvalue $2\pi/e^T Fe$. Again, this vector corresponds to a support vector with samples from a circle. The support vectors $\alpha e + h_{\text{Null}}$, for any positive α, still correspond to the smoothest objects for the measure $\widehat{\mathcal{R}}$.

To find $\widetilde{\mathcal{R}}$ and $\widehat{\mathcal{R}}$ in the 3-dimensional case the quantity f_i, the area of a triangular face, must be found. If the sides of such a triangle are of length a, b, and c then the corresponding area is given by $1/4\sqrt{2a^2b^2 + 2b^2c^2 + 2a^2c^2 - a^4 - b^4 - c^4}$. Since the vertices of the face are a linear function of the support values, the squared lengths a^2, b^2, and c^2 are *quadratic* functions of $\mathbf{h}$, and so will f_i be. Numerically finding these values is straightforward if the neighbor structure of the faces is known.

6.5.4 Use in Optimization

In an optimization problem we can imagine using a cost function which combines both the distance of our estimate from the observations and the degree of roughness of the estimate, as provided by the measures of this section, $\mathcal{R}$ or $\widehat{\mathcal{R}}$. For example we could find a support vector estimate $\hat{\mathbf{h}}$ as the solution of:

$$\hat{\mathbf{h}} = \underset{\bar{Q}\mathbf{h} \geq 0}{\arg\min} \left[(1-\alpha)\|\mathbf{h} - \mathbf{h}_{\text{obs}}\|_2^2 + \alpha \mathcal{R}(\mathbf{h}) \right] \tag{6.20}$$

where $\mathbf{h}_{\text{obs}}$ is the observed noisy support vector, the constraint $\bar{Q}\mathbf{h} \geq 0$ imposes consistency on the estimate, and $0 \leq \alpha < 1$ is a parameter which allows a tradeoff between adherence to data and smoothness. Note that if $\alpha = 1$ there is no unique minimum to the problem, since scalar multiples of a support vector have the same cost. As α approaches 1, however, we would expect the resulting estimate to approach the unique polyhedron surrounding a sphere which is closest to the observations in average support deviation. This is indeed what is observed in practice.

To illustrate these points for the planar case, we show such a family of estimates from noisy support data in Figure 6-31. In the upper left of the figure the noisy inconsistent support data is displayed. The upper right shows the closest consistent estimated support vector, corresponding to the solution of (6.20) with $\alpha = 0$. The reconstructed support vector for this choice of α produces an oblong object. In the

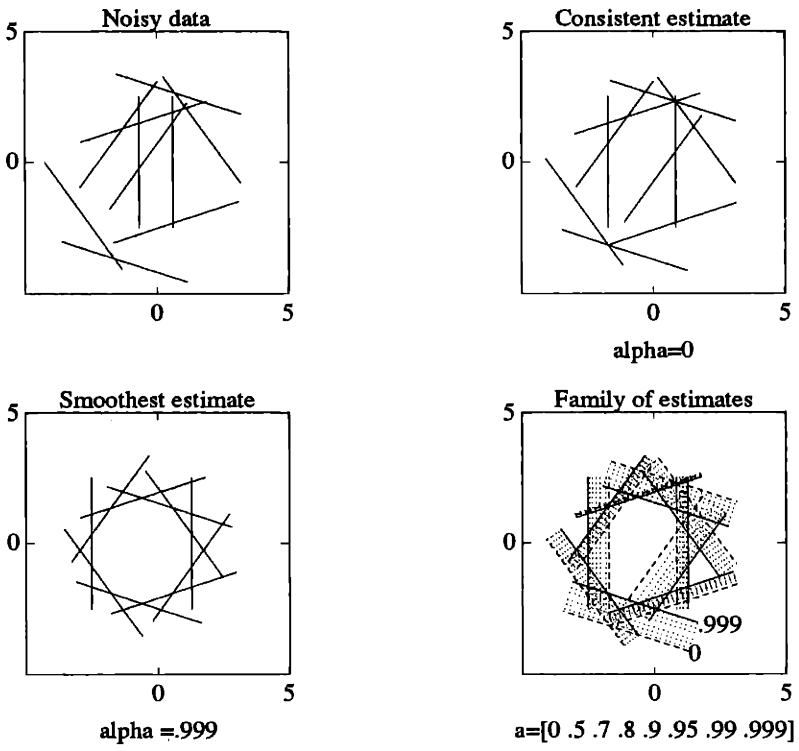

Figure 6-31: Family of support reconstructions.

bottom left is the support reconstruction corresponding to letting $\alpha \to 1$ in (6.20), yielding the smoothest estimate from this data. The eccentricity in the reconstruction has been removed and the samples surround a circle. Finally, in the bottom right hand corner of Figure 6-31 a family of reconstructions is shown for various values of α between the extremes of 0 and 1. The extreme reconstruction corresponding to $\alpha = 0$ is shown as dashed lines and that for $\alpha \to 1$ is given by solid lines. Other members of the family are shown as dotted lines. The values of α used for the reconstructions are given along the bottom of the plot.

6.6 Summary and Comments

In this chapter we have investigated issues arising from the use of discrete support measurements (or samples) in the reconstruction of objects. We began with a review and summary of classical results and relations involving the support *functions* of objects. The insight developed here formed the basis of our extensions to the discrete case. One such classical result due to Rademacher was generalized from the planar

case to the case of arbitrary dimensions. We then used this result to extend the planar, equal-angle support sample consistency results of [8] to this general setting. This generalization also used our interpretation of the planar results of [8] as a series of *local* consistency tests, a notion which we carried to the higher dimensional case.

The requirements imposed by a *smoothness* constraint were then considered. First we approached this constraint *locally* by analyzing and developing different notions of support-based discrete curvature. One group of curvature definitions were extensions of our planar results while an additional group were based directly on the geometric definition of Gaussian curvature. Convergence analysis of these definitions was given, suggesting the desirability of some over others. Their use in optimization was also briefly discussed.

In contrast to the above inherently local concept of smoothness based on surface curvature, we also presented global notions of smoothness based on the classical isoperimetric inequality. These definitions measure the deviation of an object from a sphere, the smoothest object in some sense. Our discrete definitions measure the deviation of the resulting polyhedron from a sphere and are minimized for polyhedra surrounding a sphere. For a fixed number of faces and orientations, such a polyhedron is the smoothest in this sense. These notions of smoothness were also conveniently expressed in terms of the support samples and area of the discrete object. An example of support reconstruction incorporating such a measure was given.

Many of the different curvature and smoothness notions presented in this chapter rely on the surface area (or boundary length in the planar case) f_i of the polygonal representation obtained from a given set of support measurements. While using the surface area of such polyhedral models can be tricky in general, yielding strange limiting results, our use of the area should be insensitive to such problems. To demonstrate the potential difficulty, consider the triangulated cylinder of unit height shown in Figure 6-32 from [73]. The cylinder is triangulated by dividing its height into k equal parts and its circumference into ℓ equal parts to yield a total of $2k\ell$ triangles. The overall area f_{cyl} of the triangulated surface is given in [73, pg. 597] as:

$$f_{\text{cyl}} = 2\ell \sin(\pi/\ell)\sqrt{1 + 4k^2 \sin^4(\pi/2\ell)}.$$

Now the limiting area depends strongly on how the triangulation is refined. Suppose

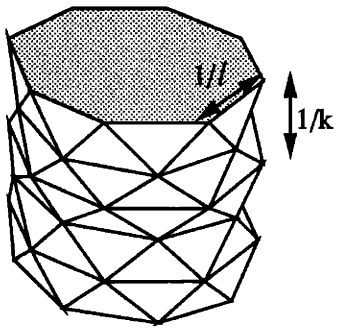

Figure 6-32: Triangulated cylinder.

we let $k = \ell^s$, then we obtain the following limits:

$$\lim_{k\to\infty} f_{\text{cyl}} = \begin{cases} 2\pi & s = 1 \\ 2\pi\sqrt{1+\pi^4/4} & s = 2 \\ \text{Does not exist} & s = 3 \end{cases}$$

Thus the triangulated area does not seem a particularly robust measure of the actual area of the cylinder. It is pointed out in [73] that the area of the cylinder is the *limes inferior* of the triangulated area, and thus forms a lower bound for it.

What this argument seems to indicate is that we should be careful about expecting to extract true values of the surface curvature from any of the discrete definitions. On the other hand this should not pose a serious problem in using such discrete quantities as a smoothness measure, since the goal is to obtain a *relative* indication of curvature.

6-A Proof of Result 28

To prove the result we need only show that condition 3′ of Result 28 implies and is implied by condition 3 of Result 27. Without loss of generality we may assume in what follows that u_{n+1} is in the positive cone of the remaining u_i. If this is not the case, we need only interchange rows in 3′ and relabel. Such operations do not change the sign of the left hand side because the row exchanges will take place in both the determinant terms. Since u_{n+1} is in the positive cone of $\{u_i\}$, $i = 1, \ldots, n$, we may write it as the following linear combination:

$$u_{n+1} = [u_1|\cdots|u_n] \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \tag{6.21}$$

with $0 \leq \alpha_i \leq 1$.

We shall repeatedly use the following result, which we state as a lemma:

Lemma 8 *Suppose that the unit vector u_{n+1} is in the full positive cone of the set $\{u_i\}$, $i = 1, \ldots, n$. Then the following equality holds for some $\beta(u_i) \geq 0$:*

$$\begin{vmatrix} H(u_1) & u_1^T \\ H(u_2) & u_2^T \\ \vdots & \vdots \\ H(u_{n+1}) & u_{n+1}^T \end{vmatrix} \begin{vmatrix} 1 & u_1^T \\ 1 & u_2^T \\ \vdots & \vdots \\ 1 & u_{n+1}^T \end{vmatrix} = \beta(u_i) \left(u_{n+1}^T \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_n^T \end{bmatrix}^{-1} \begin{bmatrix} H(u_1) \\ H(u_2) \\ \vdots \\ H(u_n) \end{bmatrix} - H(u_{n+1}) \right) \tag{6.22}$$

Proof of lemma: Apply the following determinantal identity to each term of the left hand side above:

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = |A||D - CA^{-1}B|$$

Doing this and equating terms above shows that the scalar β is given by:

$$\beta(u_i) = \begin{vmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_n^T \end{vmatrix}^2 \left(u_{n+1}^T \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_n^T \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} - 1 \right)$$

Now the first term of β is clearly nonnegative. Substituting the expression (6.21) for u_{i+1} into the second term shows that it is equal to $(\sum_{i=1}^{n} \alpha_i - 1)$. Since u_{n+1} is a unit vector we have:

$$\begin{aligned} 1 &= \|u_{n+1}\| = \sum_{i=1}^{n} \alpha_i + 2 \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \alpha_i^2 \alpha_j u_i^T u_j \\ &\leq \sum_{i=1}^{n} \alpha_i^2 + 2 \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \alpha_i \alpha_j = \left(\sum_{i=1}^{n} \alpha_i \right)^2 \end{aligned}$$

so that $\sum \alpha_i \geq 1$. Thus this term is nonnegative also and the lemma is shown.

Now we show that 3 implies 3'. Combining (6.22) of the lemma above with (6.21) we obtain:

$$\begin{vmatrix} H(u_1) & u_1^T \\ H(u_2) & u_2^T \\ \vdots & \vdots \\ H(u_{n+1}) & u_{n+1}^T \end{vmatrix} \begin{vmatrix} 1 & u_1^T \\ 1 & u_2^T \\ \vdots & \vdots \\ 1 & u_{n+1}^T \end{vmatrix} = \beta(u_i) \left(\sum_{i=1}^{n} \alpha_i H(u_i) - H(\sum_{i=1}^{n} \alpha_i u_i) \right)$$

with $\beta(u_i) \geq 0$. Since the subadditivity condition 3 holds, $H(\alpha_i u_i + \alpha_j u_j) \leq H(\alpha_i u_i) + H(\alpha_j u_j)$ and we have that

$$\begin{aligned} \begin{vmatrix} H(u_1) & u_1^T \\ H(u_2) & u_2^T \\ \vdots & \vdots \\ H(u_{n+1}) & u_{n+1}^T \end{vmatrix} \begin{vmatrix} 1 & u_1^T \\ 1 & u_2^T \\ \vdots & \vdots \\ 1 & u_{n+1}^T \end{vmatrix} &= \beta(u_i) \left(\sum_{i=1}^{n} H(\alpha_i u_i) - H\left(\sum_{i=1}^{n} \alpha_i u_i \right) \right) \\ &\geq 0 \end{aligned}$$

and thus condition 3 implies condition 3'.

Now we show that condition 3' implies condition 3. Given arbitrary vectors v and w, we will show that if 3' is satisfied then $H(v+w) \leq H(v) + H(w)$. If v is a scalar multiple of w this is trivially true from condition 2 or 2'. Assume such is not the case. In condition 3' let $u_1 = v/\|v\|$, $u_2 = w/\|w\|$ and choose the remaining u_i, $i = 3, \ldots, n$ arbitrarily to span the subspace perpendicular to v and w. Let $u_{n+1} = (v+w)/\|v+w\|$, so that u_{n+1} is a unit vector in the full positive cone of the

u_i. Then we have that:

$$u_{n+1} = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix} \begin{bmatrix} \frac{\|v\|}{\|v+w\|} \\ \frac{\|w\|}{\|v+w\|} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Now by condition 3′ and (6.22) of the lemma it follows that

$$u_{n+1}^T \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_n^T \end{bmatrix}^{-1} \begin{bmatrix} H(u_1) \\ H(u_2) \\ \vdots \\ H(u_n) \end{bmatrix} - H(u_{n+1}) \geq 0$$

for unit vectors u_i, with u_{n+1} in the positive cone of $\{u_i\}$, $i = 1, \ldots, n$. Substituting the expressions above for the u_i and u_{n+1} we obtain

$$\begin{bmatrix} \frac{\|v\|}{\|v+w\|} & \frac{\|w\|}{\|v+w\|} & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_n^T \end{bmatrix} \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_n^T \end{bmatrix}^{-1} \begin{bmatrix} H(u_1) \\ H(u_2) \\ \vdots \\ H(u_n) \end{bmatrix} - H\left(\frac{v+w}{\|v+w\|}\right) =$$

$$\frac{\|v\|}{\|v+w\|} H\left(\frac{v}{\|v\|}\right) + \frac{\|w\|}{\|v+w\|} H\left(\frac{w}{\|w\|}\right) - H\left(\frac{v+w}{\|v+w\|}\right) \geq 0.$$

Equivalently, using condition 2,

$$\frac{1}{\|v+w\|} \left(H(v) + H(w) - H(v+w)\right) \geq 0.$$

Thus

$$H(v+w) \leq H(v) + H(w)$$

and the converse is shown. Together these implications prove the result.

6-B Proof of Result 29

We show that under the hypotheses of the result the enlarged set $\{h_1, \ldots, h_n, h_{n+1}\}$ is consistent. Consistency of the set $\{h_1, \ldots, h_n, h_{n+2}\}$ then follows by symmetry. We know that $u_{n+1} \in \mathsf{cone}^+\{u_1, \ldots, u_{n-1}, u_{n+2}\}$ and $u_{n+2} \in \mathsf{cone}^+\{u_2, \ldots, u_n, u_{n+1}\}$, thus we may write u_{n+1} and u_{n+2} as the following linear combinations:

$$u_{n+1} = \begin{bmatrix} u_1 & u_2 & \cdots & u_{n-1} & u_n \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n-1} \\ 0 \end{bmatrix} + \alpha_{n+2} u_{n+2} \tag{6.23}$$

$$u_{n+2} = \begin{bmatrix} u_1 & u_2 & u_3 & \cdots & u_n \end{bmatrix} \begin{bmatrix} 0 \\ \bar{\alpha}_2 \\ \bar{\alpha}_3 \\ \vdots \\ \bar{\alpha}_n \end{bmatrix} + \bar{\alpha}_{n+1} u_{n+1} \tag{6.24}$$

where $0 \leq \alpha_i \leq 1$ and $0 \leq \bar{\alpha}_i \leq 1$. Note that α_n and $\bar{\alpha}_1$ are 0. We may eliminate u_{n+2} from the above two expressions to obtain the following equivalent expression for u_{n+1}:

$$u_{n+1} = \frac{1}{1 - \bar{\alpha}_{n+1}\alpha_{n+2}} \begin{bmatrix} u_1 & u_2 & \cdots & u_{n-1} & u_n \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 + \bar{\alpha}_2\alpha_{n+2} \\ \vdots \\ \alpha_{n-1} + \bar{\alpha}_{n-1}\alpha_{n+2} \\ \bar{\alpha}_n\alpha_{n+2} \end{bmatrix} \tag{6.25}$$

where $\bar{\alpha}_{n+1}\alpha_{n+2} < 1$ since u_{n+1} and u_{n+2} are distinct by assumption.

Using (6.8) and Lemma 8, consistency of the set $\{h_1, \ldots, h_{n-1}, h_{n+1}, h_{n+2}\}$ implies that the following inequality is satisfied:

$$u_{n+1}^T \begin{bmatrix} u_1 & u_2 & \cdots & u_{n-1} & u_{n+2} \end{bmatrix}^{-T} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n-1} \\ h_{n+2} \end{bmatrix} - h_{n+1} \geq 0$$

Substitution of (6.23) for u_{n+1} into this inequality and rearrangement yields

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_{n-1} & 0 & -1 & \alpha_{n+2} \end{bmatrix} \mathbf{h} \geq 0 \tag{6.26}$$

where $\mathbf{h} = [h_1, h_2, \cdots, h_n, h_{n+1}, h_{n+2}]^T$ is the support vector. Similarly, using (6.8) and Lemma 8 together with (6.24), consistency of the set $\{h_2, \ldots, h_n, h_{n+1}, h_{n+2}\}$ yields the following inequality:

$$[\ 0 \quad \bar{\alpha}_2 \quad \bar{\alpha}_3 \quad \cdots \quad \bar{\alpha}_n \quad \bar{\alpha}_{n+1} \quad -1\ ]\mathbf{h} \geq 0 \tag{6.27}$$

Thus, (6.26) and (6.27) are satisfied by assumption.

To show consistency of the samples $\{h_1, \ldots, h_n, h_{n+1}\}$ we have to show that the following expression is nonnegative

$$\begin{vmatrix} h_1 & u_1^T \\ h_2 & u_2^T \\ \vdots & \vdots \\ h_{n+1} & u_{n+1}^T \end{vmatrix} \begin{vmatrix} 1 & u_1^T \\ 1 & u_2^T \\ \vdots & \vdots \\ 1 & u_{n+1}^T \end{vmatrix}$$

Applying Lemma 8 again and substituting for u_{n+1} from (6.25) shows that this expression is equivalent to

$$\frac{\beta}{1 - \bar{\alpha}_{n+1}\alpha_{n+2}}[\ \alpha_1 \quad (\alpha_2 + \bar{\alpha}_2\alpha_{n+2}) \quad (\alpha_3 + \bar{\alpha}_3\alpha_{n+2}) \quad \cdots \quad (\alpha_{n-1} + \bar{\alpha}_{n-1}\alpha_{n+2}) \quad \bar{\alpha}_n\alpha_{n+2} \quad (\bar{\alpha}_{n+1}\alpha_{n+2} - 1) \quad 0\ ]\mathbf{h} \tag{6.28}$$

where $\beta = \beta(u_i)$ is a nonnegative scalar depending on the u_i. We may equivalently write this as

$$\frac{\beta}{1 - \bar{\alpha}_{n+1}\alpha_{n+2}}\mathbf{h}^T \left(\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{n-1} \\ 0 \\ -1 \\ \alpha_{n+2} \end{bmatrix} + \alpha_{n+2} \begin{bmatrix} 0 \\ \bar{\alpha}_2 \\ \bar{\alpha}_3 \\ \vdots \\ \bar{\alpha}_n \\ \bar{\alpha}_{n+1} \\ -1 \end{bmatrix} \right).$$

which will be recognized as a linear combination of the left hand sides of (6.26) and (6.27). Now the terms $\beta/(1 - \bar{\alpha}_{n+1}\alpha_{n+2})$ and α_{n+2} are nonnegative so (6.28) is equal to a nonnegative linear combination of (6.26) and (6.27), which are also nonnegative. Consequently, (6.28) must also be nonnegative and we have demonstrated consistency of the set $\{h_1, \ldots, h_n, h_{n+1}\}$. The result is thus shown.

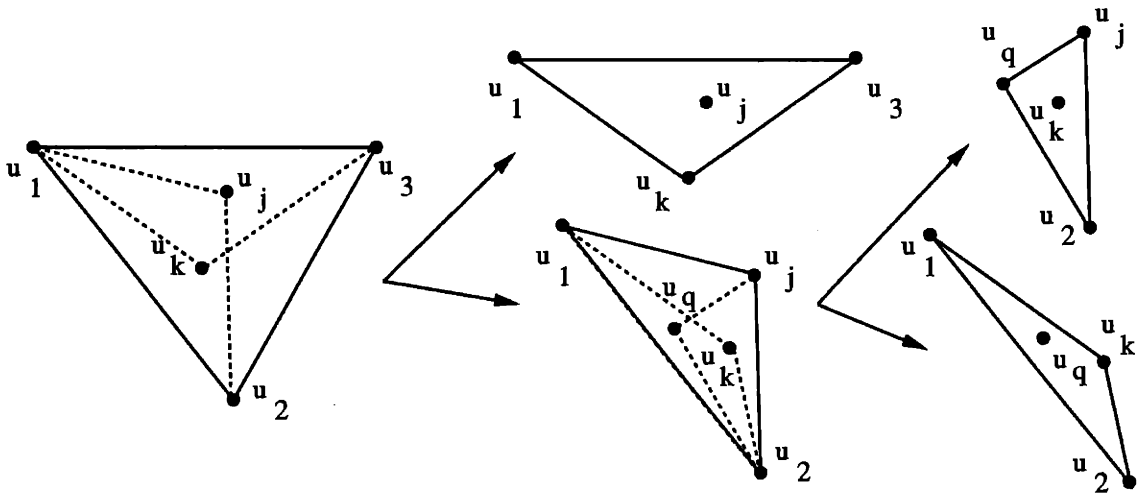

Figure 6-33: Illustration of geometry on Gaussian sphere.

6-C Proof of Result 30

That global consistency implies local consistency follows easily, for if a set is globally consistent then by definition the inequality (6.8) is satisfied for all unit vectors in the full positive cone of an n-tuple of other normals.

To show that local sample consistency implies global sample consistency we want to show that for every sample normal in the full positive cone of n other sample normals, these support samples and normals satisfy the determinantal inequality (6.8). To this end, consider an arbitrary support sample h_j and its associated unit normal u_j, in the positive cone of some (possibility non-local) n-tuple of other sample normals. On the surface of the n-dimensional Gaussian spheroid this unit vector (in the positive cone of n other unit vectors) is a point inside an $(n-1)$-dimensional spherical simplex (generalization of a spherical triangle), as described in association with Figure 6-11 for the 3-dimensional case. Points in this simplex are points in the positive cone of the vectors at the vertices of the simplex, as is u_4 in the figure.

If the point associated with u_j is isolated in the simplex it is consistent by hypothesis, since it is the only vector in the positive cone of the vertex normals, and hence part of a local family. Suppose instead that there is another point in the simplex with it, say u_k as shown in the leftmost illustration of Figure 6-33 for the 3-dimensional case. Each of these two interior points in combination with the n original bounding points tessellates the original simplex into n disjoint subsimplicies. The two interior points u_j and u_k are thus each contained in a subsimplex formed from the other inte-

rior point and $n-1$ of the original boundary points. This geometry is shown in the leftmost frame of the figure as two dotted triangles.

Now by the merging result, Result 29, if we can show that the samples u_j and u_k are consistent on these smaller subregions then we have shown that they are consistent on the entire region, as desired. Thus we have reduced the problem from showing consistency over the original region to showing consistency over two smaller subregions. This process is shown in the middle illustration of the figure, where we have split the original test into two subtests. We may now repeat the above arguments on each of the subregions, attempting to show the consistency of each. We thus have the following finitely terminating recursive construction:

Show an arbitrary $(n+1)$-tuple is consistent:

1. If it is isolated in its cone, consistency is shown by hypothesis.

2. If it is not isolated:

   (a) Pick another point in the cone.

   (b) Form two smaller subregions.

   (c) Show consistency of each subregion.

We keep proceeding in this way until a subregion is found where the interior point is isolated (corresponding to the normal being the only vector in the positive cone of its bounding set) and consistency is satisfied by hypothesis. In Figure 6-33 we show the next step of this procedure on the right, where we have assumed that the subregion containing u_j is isolated (so the we have reached a leaf of the tree) but the one containing u_k contains another point, u_q and hence must itself be broken into two subregions.

Since at each stage of this procedure another point (sample normal) is removed from the original finite set and since the subregions are nonincreasing at each stage, we must eventually reach the situation where a sample normal is isolated in its simplex and therefore consistent. We may then travel back up the tree we have implicitly created to show consistency of the original normal with respect to the original boundary points. Since the normal and surrounding cone we chose were arbitrary, we have shown the result.

6-D Derivation of Curvature Limits

In this appendix we derive the various limiting expressions for the different curvature expressions of section 6.4.

6-D.1 Definition $\mathcal{K}_1$

Refering to Figure 6-16 we find that

$$f_i = \frac{x_{i+1} - x_{i-1}}{2} = \frac{\tan(\Delta\theta_i) + \tan(\Delta\theta_{i+1})}{2g}$$

where the second equality follows from the fact that $x_{i+1} = \tan(\Delta\theta_{i+1})/g$ and $x_{i-1} = -\tan(\Delta\theta_i)/g$. Thus we obtain for $\mathcal{K}_1(i)$:

$$\mathcal{K}_1(i) \equiv \frac{\Delta\theta_i}{f_i} = \frac{2g\Delta\theta_i}{\tan(\Delta\theta_i) + \tan(\Delta\theta_{i+1})}$$

Now for convergence analysis let $\Delta\theta_{i+1} = \alpha\Delta\theta_i$ so that the scalar α measures the realtive rate of convergence of the two angles. Letting $\Delta\theta_i$ go to zero yields for the limiting curvature:

$$\lim_{\Delta\theta_i \to 0} \mathcal{K}_1(i) = \lim_{\Delta\theta_i \to 0} \frac{2g}{\sec^2(\Delta\theta_i) + \alpha \sec^2(\alpha\Delta\theta_i)} = 2g\frac{1}{1+\alpha}.$$

Thus the proper curvature is obtained only if $\alpha = 1$ so that convergence is uniform.

6-D.2 Definition $\mathcal{K}_2$

Using the expression derived in the previous section for f_i and in definition of $\mathcal{K}_2$ we obtain

$$\mathcal{K}_2 \equiv \frac{\Delta\theta_i + \Delta\theta_{i+1}}{2f_i} = \frac{g(\Delta\theta_i + \Delta\theta_{i+1})}{\tan(\Delta\theta_i) + \tan(\Delta\theta_{i+1})}$$

Again for convergence analysis let $\Delta\theta_{i+1} = \alpha\Delta\theta_i$ for a scalar α measuring the realtive rate of convergence of the two angles. Letting $\Delta\theta_i$ go to zero yields for the limiting

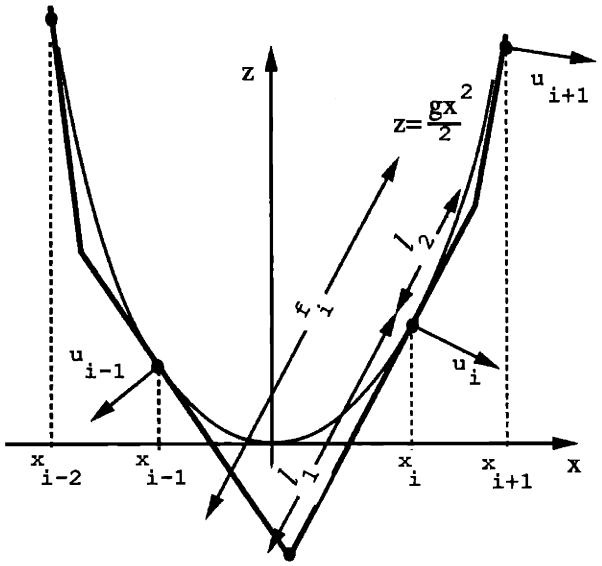

Figure 6-34: Finding f_i for $\mathcal{K}_3$.

curvature:

$$\lim_{\Delta\theta_i \to 0} \mathcal{K}_1(i) = \lim_{\Delta\theta_i \to 0} \frac{g(1+\alpha)}{\sec^2(\Delta\theta_i) + \alpha \sec^2(\alpha \Delta\theta_i)} = g.$$

Thus in this case the proper curvature is obtained for any α, i.e. for any relative convergence of $\Delta\theta_i$ and $\Delta\theta_{i+1}$.

6-D.3 Definition $\mathcal{K}_3$

First we shall find $\Delta\theta_i$ in terms of the x_i. The unit normal u_i is given by

$$u_i = \frac{1}{\sqrt{g^2 x_i^2 + 1}} [g x_i, -1]^T$$

and similarly for u_{i-1}. Since $\cos(\Delta\theta_i) = u_{i-1}^T u_i$, we obtain for $\Delta\theta_i$

$$\Delta\theta_i = \arctan\left(\frac{g(x_i - x_{i-1})}{g^2 x_i x_{i-1} + 1}\right).$$

Now to find f_i consider Figure 6-34. The point of intersection of support line $i-1$ and i is given by $[(x_{i-1} + x_i)/2, g x_{i-1} x_i / 2]^T$. Thus we obtain for the partial lengths

ℓ_1 and ℓ_2 indicated in the figure

$$\begin{aligned} \ell_1 &= \left(\frac{x_i - x_{i-1}}{2}\right)^2 \sqrt{g^2 x_i^2 - 1} \\ \ell_2 &= \left(\frac{x_{i+1} - x_i}{2}\right)^2 \sqrt{g^2 x_i^2 - 1}. \end{aligned}$$

The length f_i is given by the sum of these two components. Performing a similar proceedure for f_{i-1} yields:

$$\begin{aligned} f_i &= \left(\frac{x_{i+1} - x_{i-1}}{2}\right)^2 \sqrt{g^2 x_i^2 - 1} \\ f_{i-1} &= \left(\frac{x_i - x_{i-2}}{2}\right)^2 \sqrt{g^2 x_{i-1}^2 - 1} \end{aligned}$$

Now applying the definition of $\mathcal{K}_3(i)$ we obtain:

$$\mathcal{K}_3(i) = \frac{4 \arctan\left(\frac{g(x_i - x_{i-1})}{g^2 x_i x_{i-1} + 1}\right)}{(x_{i+1} - x_i)\sqrt{(g^2 x_i^2 + 1)} + (x_i - x_{i-2})\sqrt{(g^2 x_{i-1}^2 + 1)}}$$

where we have expressed the curvature as a function of the x coordinates of the tangency points to the curve x_{i-2}, x_{i-1}, x_i, x_{i+1}.

This time for the convergence analysis we work with the x coordinates of the tangency points. Let the following relationships exist between the different x coordinates:

$$\begin{aligned} \alpha x_{i+1} &= x_i \quad 0 \leq \alpha \leq 1 \\ x_{i-1} &= -\beta x_i \quad 0 \leq \beta \\ \gamma x_{i-2} &= x_{i-1} \quad 0 \leq \gamma \leq 1 \end{aligned}$$

so that the relative places of the x coordinates is maintained. Substituting these expressions for x_i into the formula for $\mathcal{K}_3(i)$ and letting x_i go to zero we obtain for the limiting curvature:

$$\lim_{x_i \to 0} \mathcal{K}_3(i) = g \frac{4\alpha\gamma(\beta + 1)}{\alpha\beta\gamma + \alpha\beta + \alpha\gamma + \gamma}.$$

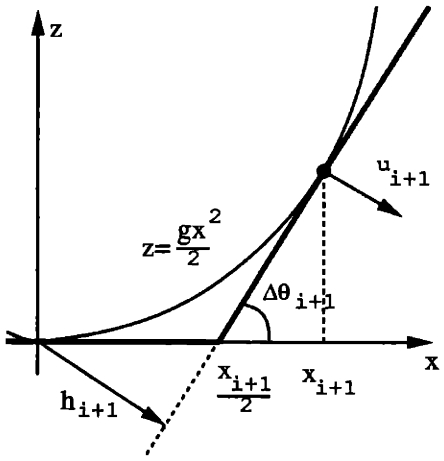

Figure 6-35: Finding h_{i+1} for $\mathcal{K}_4$.

For this value to equal the underlying value of g we require

$$\frac{4\alpha\gamma(\beta+1)}{\alpha\beta\gamma+\alpha\beta+\alpha\gamma+\gamma}=1.$$

6-D.4 Definition $\mathcal{K}_4$

First we shall find the support values h_{k-1} and h_{k+1} as a function of angle increments $\Delta\theta_i$ and $\Delta\theta_{i+1}$. Note that because of the location of the origin the support value h_i is 0. Refer to Figure 6-35 in what follows. From simple geometry we have that:

$$\tan(\Delta\theta_{i+1})=gx_{i+1}$$

and that

$$h_{i+1}=\frac{x_{i+1}}{2}\sin(\Delta\theta_{i+1}).$$

Combining these two we obtain for h_{i+1}:

$$h_{i+1}=\frac{\sin^2(\Delta\theta_{i+1})}{2g\cos(\Delta\theta_{i+1})}$$

Similarly we get for h_{k-1} the expression

$$h_{i-1} = \frac{\sin^2(\Delta\theta_i)}{2g\cos(\Delta\theta_i)}$$

Now combining these two expressions with $h_i = 0$ in the definition of $\mathcal{K}_4(i)$ and simplifying the result yields the following expression for curvature of the parabola at the origin:

$$\mathcal{K}_4(i) = \frac{2g\Delta\theta_i\Delta\theta_{i+1}\cos(\Delta\theta_i)\cos(\Delta\theta_{i+1})}{\cos(\Delta\theta_i)\sin^2(\Delta\theta_{i+1}) + \cos(\Delta\theta_{i+1})\sin^2(\Delta\theta_i)}.$$

For convergence analysis let $\Delta\theta_{i+1} = \alpha\Delta\theta_i$. Substituting this relationship into the curvature expression $\mathcal{K}_4(i)$ above and letting $\Delta\theta_i$ go to 0 we obtain for the limiting curvature:

$$\lim_{\Delta\theta_i \to 0} \mathcal{K}_4(i) = g\frac{2\alpha}{\alpha^2 + 1}$$

where two applications of L'hopital's rule are required to yield the above expression. This limiting value will equal the underlying value of g if and only if $\alpha = 1$.

Chapter 7

Further Work

In this chapter we present some ideas and problems that seem like fruitful avenues for future work. First we discuss ideas closely related to the body of this thesis and the issues raised therein. Then some problems more distantly linked to the present work will be presented. Partial results for these problems are provided to illustrate potential methods and to encourage their further examination.

7.1 Thesis Related Work

In this section we present some problems for further consideration which are directly related to the work presented in this thesis. Throughout this thesis we have called attention to many such issues as they arose. Here we collect some of these and suggest some new ones. We examine these problems chapter by chapter.

7.1.1 Symmetric Matrices

One open area for work from Chapter 2 concerns our approximations of the PSD cone in Section 2.4. Further development of measures to determine the distance of an approximating set from the exact PSD cone is needed. In Section 2.4.2 we provided one such measure based on the angle between certain extreme elements of the approximation and the PSD set. This measure was only valid for support based exterior approximations. A convenient way to compare all PSD set approximations

would be valuable. Part of the problem is that the PSD set is not compact so that set measures like the Hausdorff measure are infinite. An associated extension involves relating the approximating sets and the reconstructions based on them. It would be desirable to be able to bound the maximum error of an estimate based on a particular PSD approximation. This task is harder than the previous one of simply comparing the distance of set approximations from the PSD set.

Another area for exploration is the relationship between our problems and results and the work on *M-matrices*. A real matrix is termed an M-matrix if it has negative off-diagonal elements and all its principal minors are positive [112]. Symmetric M-matrices are positive definite. Thus symmetric M-matrices are valid representations of both curvature and ellipsoids. Conversely, our techniques for estimation of a PSD matrix may be of use in problems involving M-matrices.

Finally, a host of other problems appear to fit within the framework developed in this chapter. For example, the inverse conductivity problem of determining the symmetric conductance matrix of a network from a series of current or voltage probes results in an equation that seems to be in the form the projection mapping (2.3). The medical imaging method of electrophoresis, whereby voltage probes at the body's surface are used to determine resistivity distribution, fits this mold also.

7.1.2 Matrix Reconstruction

One possible way to extend the results of Chapter 3 is to use our constrained symmetric matrix reconstruction algorithms to perform constrained reconstruction of more general matrices. For example, using a Cartesian decomposition we may write any matrix X as $X = X_{\text{symm}} + X_{\text{skew}}$, where $X_{\text{symm}} = (X + X^T)/2$ is the symmetric part and $X_{\text{skew}} = (X - X^T)/2$ is the skew-symmetric part. From Rayleigh quotient arguments the real part of the eigenvalues of X are bounded by the (real) eigenvalues of X_{symm} and the imaginary part of the eigenvalues of X are bounded by the eigenvalues of X_{skew}. We might desire to estimate a matrix with bounds on the real parts of its eigenvalues. This could be done by separately estimating its symmetric and skew-symmetric parts. The reconstruction of the symmetric part could then make use of the constrained algorithms of Chapter 3. It is unclear exactly how to incorporate constraints on the skew-symmetric part for the case of real matrices. For matrices over the complex field, an isomorphism exists between what are termed Hermitian and skew-Hermitian matrices. Such bounded eigenvalue problems are of wide interest

in many fields, particularly as they pertain to stability.

Another potential area for investigation is raised by the successive halfspace algorithm of Section 3.3.3. This algorithm solves a specially structured, linear inequality constrained, least squares problem (LSI) at each stage. While the constraint set grows from iteration to iteration, the previous constraints at a given stage remain *unchanged.* Thus some sort of recursive form of LSI would be highly desirable. While it seems a if there should be some way to use previous solutions to aid the current estimate, this could be a very hard problem. One general difficulty is that as more constraints are added some may no longer be binding, and as we discussed in Section 6.3.1, determining which constraints are binding essentially requires solving the problem. Our particular problem has special structure however. Since the constraints correspond to support halfspaces of a convex cone and they are all different, they *must* all be binding from geometric considerations (they will of course not all be *active* however). Perhaps a recursive form of LSI may be found for this special case.

Another way to reduce the computational burden of the successive halfspace approach of Algorithm 1 is to replace the eigenanalysis at each iteration by approximate expressions. In the limit, we may assume that the estimate does not change much from iteration to iteration. As a result we could use first order perturbation expressions for the eigenvalues and eigenvectors and treat the change in the estimate from iteration to iteration as a perturbation. Since the algorithm seems to converge quickly, we would expect such an approach to work well. The computational savings would be especially great on larger problems

7.1.3 Curvature

Certainly one area for investigation from Chapter 4 is suggested by the Gaussian sphere interpretation of the curvature-based global shape reconstruction problem discussed in Section 4.3.4. Specially, the question is how to best use the limited information obtained from any finite set of projections to reconstruct an overall shape. There would seem to be a relationship between the smoothness of the surface and the corresponding need for samples. For example, if we knew the object was a sphere (the smoothest of objects is some sense) then the determination of surface curvature at a single point is all that is necessary to recover the shape. In general, we would expect that smoother objects would require fewer curvature samples to be uniquely determined. Such issues seem related to some type of surface curvature Nyquist criterion.

The challenge is to make such notions precise.

7.1.4 Ellipsoids

There are a number of interesting directions to proceed in from the work of Chapter 5. One important extension involves the application of our results to non-centered ellipsoids. We may of course simply translate the coordinate system to obtain results in the case of simple shifts of the elements. More difficult is the extension of the ellipsoidal bounds, corresponding to an interval matrix set, to the case of non-centered bounding ellipsoids with different centers. This case does not directly correspond to an interval matrix set any more due to the differing shifts in the centers.

Another interesting area involves the dynamic formalism of Section 5.5.2. Further investigation of robustness issues pertaining to such dynamically oriented reconstructions are needed. Related to this is the application of the consistency results of Chapter 6 to this dynamic setting. Such an application was demonstrated for the static case through the use of a 2-step reconstruction approach, with support consistency imposed in the first step. Similar approaches should be possible for the dynamic case, presumably with good results for the high noise case. We briefly discussed such issues in Section 5.5.2.

Finally, our dynamic work in this chapter assumed knowledge of the system. An interesting problem is raised by the possibility of identifying the dynamic system based on the observations. The topic of system identification has received much attention in general and our linear formulation should be amenable to the methods generally used there.

7.1.5 Support Measurements

In Chapter 6, we presented many different discrete curvature definitions for both the 2- and 3-dimensional cases. An interesting questions, relating to our work in Chapter 4 on curvature-based reconstruction, is whether *discrete* curvature-based reconstruction notions can be defined which recover the continuous case in the limit. Such an endeavor would involve relating existing work on reconstruction of polyhedral shapes from projections to the discrete curvature definitions of Section 6.4. For example, consider a vertex on the contour generator of a projected polyhedron in 3-dimensions. Because the projection of a polyhedron results in a contour generator

which is along the edges of the polyhedron, we would expect a certain "threshold" behavior to the shadow as we rotate the polyhedron about this vertex. At certain points in the rotation some edges will enter the contour generators and others will leave. At first, reconstruction of the shape of a vertex would seem to require projections involving *all* the edges incident at the vertex. The minimum number of projections for reconstruction would then depend on the number on faces at the vertex. But this cannot be if we are to match the limiting requirement of only 3 planar projections for a smooth object. How do we reconcile this continuous limit with the polyhedral situation?

7.2 Structure from Motion

The body of this thesis was concerned with the reconstruction of continuous objects from their projections. Ellipsoids and smooth surfaces were examined in detail. In this section, we consider some issues in the reconstruction of points sets from their projections. For a rigid point set, such problems are classically known as structure-from-motion problems [113, 114, 115]. This term follows from pioneering work of Ullman, who proved that the 3-dimensional relationship of four rigidly connected, non-coplanar points (a simplex) can be uniquely determined from three independent but unknown orthogonal projections of the points [115]. The "motion" in the name comes from the assumption that between these views the object moves in some unknown way, providing a different aspect in each view. Note that this is equivalent to fixing the object and moving the camera to different (and unknown) spots for each view. Thus, the "motion" in the title is interchangeable with the phrase "different, unknown, views."

To our knowledge, results for Ullman's type of problem are restricted to reconstructing 3-dimensional objects from planar (2-dimensional) projections or 2-dimensional objects from scalar (1-dimensional) observations. Further, the arguments and proofs are very case-specific. Ullman actually used straightforward *substitution* to solve a set of generic observations for a general object [115]. Little insight is thereby provided for the general case of higher dimensions, more feature points, etc. Finally, trigonometric relations are often used in the formulations (for the projections), yielding nonlinear equations in the problem parameters. This is algorithmically inconvenient when developing solution schemes. In this section we introduce a formalism

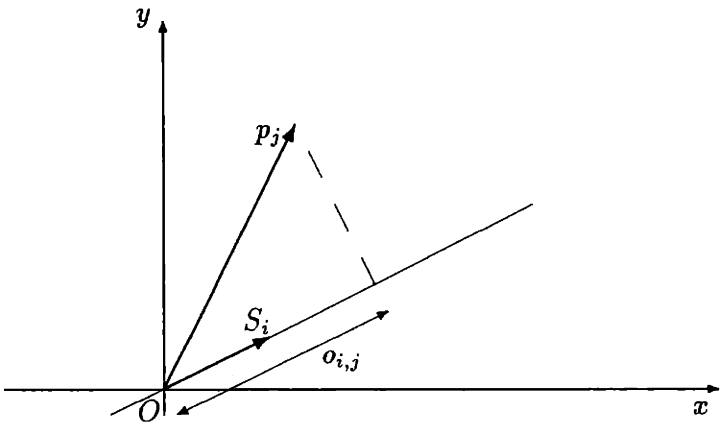

Figure 7-1: Object and view geometry

that permits us to view this problem as one in bilinear matrix estimation. This has not been done, to our knowledge, and offers potential advantages in its easy extension to arbitrary dimension and data set size, its amenability to iterative solution, and its compact separation and representation of information.

7.2.1 Fixed Object Interpretation

The situation we examine is shown schematically in Figure 7-1. Figure 7-1. An (initially rigid) object is observed in orthogonal projection. An object (assumed for now to be rigid) is observed in orthogonal projection. We first take the viewpoint that the object or point set $\mathcal{P}$ is fixed with one point at the origin and it is the *views* that move from image to image. An equivalent interpretation, which will be treated later, is that the viewpoint is fixed and the object moves. We may assume that one point of the object is fixed at the origin, since this does not change its structure. Let the columns of the $n \times m_i$ matrix S_i form an orthonormal basis for the projection subspace $\mathcal{S}_i$. Here n is the dimension of the space and m_i is the dimension of projection subspace i. For example, if the object, and hence the original space, is 3-dimensional and the observations are projections onto planes (2-dimensional), S_i will be 3×2 matrix whose columns span the plane of subspace i. Note that $S_i^T S_i = I$.

Now let the j-th token or object point be represented by an n vector from the

origin, p_j. The i-th view or observation o_{ij} of the j-th object point is then just

$$S_i^T p_j = o_{ij} \tag{7.1}$$

where o_{ij} is $m_i \times 1$. Note that o_{ij} is just a set of lengths or coordinates with respect to the (unknown) basis S_i. The matrix S_i contains all the *viewpoint* information while o_{ij} contains just the observed quantities and no viewpoint information. This formulation conveniently separates the known or observed quantities from the unknown ones. Combining terms, define the entire set of k object points (which define the object) as $\mathbf{P}$ so that

$$\mathbf{P} = [p_1|p_2|\cdots|p_k] \tag{7.2}$$

The i-th observation of the entire set of object points is then

$$S_i^T \mathbf{P} = [o_{i1}|o_{i2}|\cdots|o_{ik}] = O_i \tag{7.3}$$

Combining the separate views, we obtain an overall view matrix $\mathbf{S}$ as

$$\mathbf{S} = [S_1|S_2|\cdots|S_r]$$

Since we are viewing a rigid object, an additional constraint *between* the different S_i exists. This constraint can be seen as a requirement for the view coordinate systems to have the same handedness. Formally, view subspaces of the same dimension must be obtainable from one another just by rotations with no reflections. An equivalent requirement is that the transformation between these coordinate systems be represented by orthogonal matrices with determinant $+1$ [46, page 21]. Note also that in general, $\mathbf{S}$ will not have full column rank.

The overall set of object-view relationships can now be written as

$$\begin{bmatrix} S_1^T \\ S_2^T \\ \vdots \\ S_r^T \end{bmatrix} \mathbf{P} = \begin{bmatrix} O_1 \\ O_2 \\ \vdots \\ O_r \end{bmatrix} \tag{7.4}$$

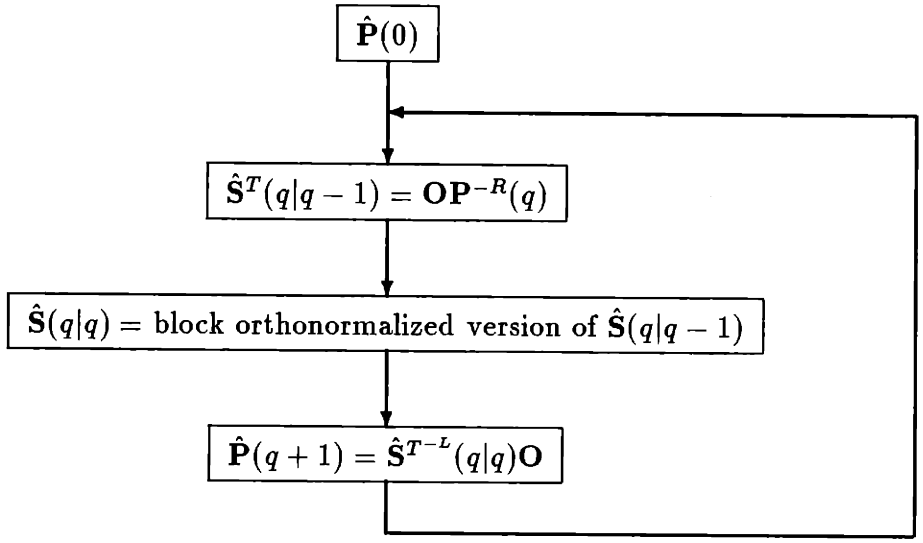

Figure 7-2: Algorithm for Ullman problem

or

$$\mathbf{S}^T\mathbf{P} = \mathbf{O} \tag{7.5}$$

where the columns of the blocks S_i are orthonormal and represent coordinate systems with the same handedness. Note that the arrangement (7.4) allows *any* combination of viewing subspace dimensions. For example, we can have a projection of the object onto a plane ($S_1 = 3 \times 2$) combined with a projection onto a line ($S_2 = 3 \times 1$) in the same problem.

For this formulation of the structure-from-motion problem, we are given the matrix $\mathbf{O}$, and the structure on $\mathbf{S}$ (i.e. that the columns of the blocks S_i are orthonormal and right handed) and asked to find both $\mathbf{S}$ and $\mathbf{P}$. Note that we may arbitrarily pick S_1 since the global reference is arbitrary. If $\mathbf{S}$ *or* $\mathbf{P}$ is given, the problem is linear in the other quantity. Thus (7.5) with the given constraints defines a *bilinear matrix estimation problem.* Further note that the formulation easily incorporates arbitrary numbers of points and views and is not biased toward either the viewpoint information or the object information.

Such bilinear estimation problems have been studied for a variety of applications, such as the work in [90] on chemical component analysis. One straightforward iterative scheme for solving (7.5) is diagramed in Figure 7-2. First a guess is made for the object $\hat{\mathbf{P}}(0)$, then this guess is used to estimate the viewpoints $\hat{\mathbf{S}}(1|0)$ using a right inverse of $\mathbf{P}$ as $\hat{\mathbf{S}}^T(1|0) = \mathbf{O}\mathbf{P}^{-R}$. Note that for a right inverse of $\mathbf{P}$ to exist we

must have the number of points greater than or equal to the dimension of the space, so $k \geq n$. This condition is satisfied for any space-filling object, such as a simplex. Now the orthonormal constraints on the columns of $\hat{S}_i$ are imposed (in some as yet unspecified fashion) to obtain $\mathbf{S}(1|1)$. A new estimate of $\hat{\mathbf{P}}(1)$ is now generated using a left inverse of $\hat{\mathbf{S}}^T(1|1)$ as $\hat{\mathbf{P}}(1) = \hat{\mathbf{S}}^{T-L}(1|1)\mathbf{O}$ and the cycle repeats. The left inverse of $\mathbf{S}^T$ only exists if the total view space dimensions equal or exceed the dimension of the space, so $\sum_{l=1}^{r} m_l \geq n$. For example, if the space is 3-dimensional, we require one 3-dimensional view or one 2-dimensional view and one 1-dimensional view or two 2-dimensional views, etc. With this iterative type approach, an estimate may be easily updated when new information (observations) become available by simply using the current guess as a starting point for iterations with the new information. Preliminary experience with this algorithm suggests it works well, converging for all examples tried.

7.2.2 Fixed View Interpretation

In the above work we assumed that the object was fixed (and rigid), and it was the *viewpoints* that moved. If all the subspaces of projection are the same dimension ($m_i = m$), then the alternate interpretation of a single, known viewpoint and a moving (rigid) object is equivalent. Again we assume (without loss of generality) that one point of the object is fixed at the origin. Let the columns of $\mathbf{S}$ form an orthonormal basis for the fixed view space and let the columns of $\mathbf{P}_i$ represent the object at its i-th viewing. For the rigid object we have assumed, the distance between the different object points will be constant so only rotations of the set are allowed. This fact constrains the $\mathbf{P}_i$. Specifically, the matrix taking $\mathbf{P}_i$ to $\mathbf{P}_j$ must be orthonormal of determinant $+1$. Now the observation of the object at view i will be:

$$\mathbf{S}^T\mathbf{P}_i = O_i \tag{7.6}$$

so the overall set of object-view relationships can be written:

$$\mathbf{S}^T [\ \mathbf{P}_1 \quad \mathbf{P}_2 \quad \cdots \quad \mathbf{P}_r\ ] = [\ O_1 \quad O_2 \quad \cdots \quad O_r\ ] \tag{7.7}$$

where the $\mathbf{P}_i$ are related as described above by orthogonal transformations of determinant $+1$. Equation (7.7) is the equivalent of (7.4) but now the viewpoint is taken as fixed and it is the object that moves between views.

We now examine this in more detail. Suppose T_i is the transformation (a rotation) taking $\mathbf{P}_i$ to $\mathbf{P}_{i+1}$ so that

$$\mathbf{P}_{i+1} = T_i \mathbf{P}_i \tag{7.8}$$

with T_i is an orthonormal matrix of determinant $+1$. Define $\mathbf{P} = \mathbf{P}_0$. Let $\tilde{T}_k \equiv \prod_{i=1}^{k} T_i$. Combining (7.7) with (7.8) we then obtain

$$\begin{bmatrix} \mathbf{S}^T \tilde{T}_1 \\ \mathbf{S}^T \tilde{T}_2 \\ \vdots \\ \mathbf{S}^T \tilde{T}_r \end{bmatrix} \mathbf{P} = \begin{bmatrix} O_1 \\ O_2 \\ \vdots \\ O_r \end{bmatrix} \tag{7.9}$$

We may also write (7.9) in the equivalent form

$$\mathbf{S}^T \begin{bmatrix} \tilde{T}_1 \mathbf{P} & \tilde{T}_2 \mathbf{P} & \cdots & \tilde{T}_r \mathbf{P} \end{bmatrix} = \begin{bmatrix} O_1 & O_2 & \cdots & O_r \end{bmatrix} \tag{7.10}$$

From these equations we can now see the relationship between our two interpretations. If the transformations $\tilde{T}_i$ are lumped with the object $\mathbf{P}$, then the case of a single fixed viewpoint and a moving object is produced. If, instead, we pair the transformations with the prototype projection subspace $\mathbf{S}^T$, the case of multiple views of a single fixed object is obtained. In this case the transformation between subspace i and subspace $i+1$ is given by T_i^T. Note that given $\mathbf{O}$ we can only determine the *relative* orientations of the different elements in the problem, and not their absolute location. For this reason we may arbitrarily assign T_1 and $\mathbf{S}$, which corresponds to choosing a particular global orientation for the problem. Even then the solution is still only unique up to a reflection due to the nature of orthogonal projection [115].

7.2.3 Dynamic Element

Even though these problems fall under the heading of structure-from-motion approaches, they really depend very little on a dynamic element. The formulation of (7.7) and (7.8) gives us a precise notion of this. From (7.8) we see that sequential positions of the object are linked by an unknown rotation. We can go a step further

and view (7.8) and (7.7) as a dynamic evolution/observation pair. The form of these equations is:

$$\begin{aligned} \mathbf{P}_{i+1} &= A_i \mathbf{P}_i \\ Y_i &= C\mathbf{P}_i \end{aligned} \tag{7.11}$$

where, for the structure-from-motion problem, $A_i = T_i$, an orthogonal matrix of determinant $+1$, $C = \mathbf{S}^T$, and $Y_i = O_i$.

Looking at the problem this way suggests other possibilities. For example, if we knew A_i in (7.11) (as we might if the rigid object were rotating at a known constant angular velocity), the problem reduces to one of straightforward state estimation. The state of the dynamic system (7.11) now represents an evolving object, however. As another possibility, consider the case when the A_i are not restricted to be rotation matrices. The dynamic equation (7.11) then models the case of identifying a *non-rigid* object from orthogonal projections. All the object points evolve the same way and any structure on A_i restricts this common evolution. We might even consider adding an input to (7.11) to drive the object motion, perhaps using this ability to identify the system [116].

7.2.4 Tomography

The formulation of (7.5) assumed that we knew the correspondence between point of the object and their projections. We may also use (7.5) to better understand a tomography like problem for point sets lacking this correspondence. In a tomography setting, $\mathbf{P}$, the object, is still desired, but now we are given $\mathbf{S}$ and the *columns* of the O_j, but with the order of these columns unknown *a priori*. This absence of column order results from the lack of object point correspondence in the tomography problem — we do not know which projection belongs to which point. Once the order of the columns is known (fixing the point correspondences), solving for $\mathbf{P}$ is easy. We may view this situation as a hypothesis testing problem with each hypothesis corresponding to a different arrangement of the columns of the O_j, and thus to a different possible set of point correspondences. The arrangement with the smallest residual is the best choice.

An iterative and recursive scheme for solving the point tomography problem above using the formulation of (7.5) again suggests itself. First, the columns of O_1 may be

assigned arbitrarily since the point labeling is not unique. Now at any stage in the estimation process there will be a best choice of column ordering. As new information (observations) become available, in the form of more blocks S_i and O_i there is no reason to change the order of the columns of the previously obtained O_j. Simply vary the ordering of the columns of the new observation. This greatly reduces the number of potential combinations to be checked. Also the scheme is recursive, combining the current best guess with the new data as it arrives.

7.2.5 Summary

In summary, the way of viewing the structure-from-motion reconstruction problem presented in this section offers a number of advantages over existing formulations. First, it nicely separates object, viewpoint, and observation information. The formulations (7.9) and (7.10) are more neutral with respect to whether the object or the views are being determined. This neutrality makes is easy to use the same equation to investigate the effects of increasing both the number of points observed and the number of views of those points. Second, this formulation exhibits a bilinear form readily amenable to iterative approaches, such as those proposed above. The goal here is to estimate the object and/or viewpoints well and not to achieve abstract sufficiency results as many efforts in the area do. Finally, extension to *any* combination of object and observation dimension and number is straightforward, giving a unified view of the general case.

7.3 Spherical Harmonics/Fourier Descriptors

The objects considered in the thesis have all been convex. One desired extension is to a less restrictive class of shapes. One such class of shapes that still possesses much structure is provided by *star-shaped* sets. A set is star-shaped with respect to a center c if, for each point p of the set, the segment c-p is also in the set, as shown in Figure 7-3. Such sets are a generalization of convex sets, where both p and c are allowed to be general elements of the set. We may represent star-shaped sets by a radius vector r from c to the set boundary as a function of direction on the unit sphere. For the planar case illustrated in Figure 7-3, for example, we would obtain $r(\theta)$. Since the function $r(\theta)$ is periodic in θ we may expand $r(\theta)$ as a Fourier series

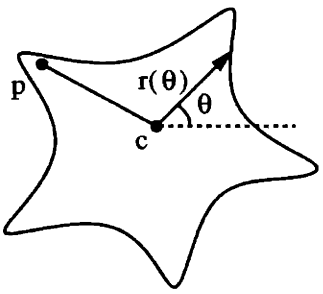

Figure 7-3: Illustration of a star-shaped set.

in θ as follows:

$$r(\theta) = \alpha_0 + \sum_{i=1}^{\infty} [\alpha_i \cos(i\theta) + \beta_i \sin(i\theta)] \tag{7.12}$$

Such a series is a representation of the boundary of the set as a sum of harmonic functions. In the higher dimensional case these functions are termed *spherical harmonics*, forming an orthonormal set of basis functions on the sphere in the same way the harmonics of sine and cosine do for the circle. The series coefficients α_i and β_i are termed the *Fourier descriptors* of the curve. The smoothness of a curve depends on the relative size of the different coefficients, smoother curves being produced by the lower order coefficients. For example, if only the coefficient α_0 is present, a circle is produced. A series with only α_0, α_1, and β_1 will only have a single bump and so forth. Such results follow easily from the more familiar results for the Fourier series of time signals.

Spherical harmonics and Fourier descriptors have been used for object recognition and classification [117, 118], for shape representation [119, 120], and for boundary filtering and analysis, particularly for biomedical applications [121, 122, 123]. In most of these schemes, the object boundary is assumed extracted in a separate step and is then converted to a finite harmonic representation for further processing. An interesting question is how we might directly use models of the form (7.12) for object reconstruction tasks.

Suppose we have access to values of the radius function r at a fixed set of m directions θ_k. In addition to the above applications where a boundary is explicitly extracted, such data might arise directly from range probes of an object at different orientations, or perhaps in a robotic setting from tactile probes. Consider a finite representation of the form (7.12) with q terms. We may collect the Fourier descriptor

coefficients in a vector as follows:

$$\mathbf{x} = [\ \alpha_0 \quad \alpha_1 \quad \beta_1 \quad \cdots \quad \alpha_q \quad \beta_q\ ]$$

Then each such sample $r_k = r(\theta_k)$ must be of the form:

$$r_k = [\ 1 \quad \cos(\theta_k) \quad \sin(\theta_k) \quad \cdots \quad \cos(q\theta_k) \quad \sin(q\theta_k)\ ]\,\mathbf{x} \tag{7.13}$$

We may stack up these observations in the following observation vector:

$$\mathbf{r} = [\ r_1 \quad r_2 \quad \cdots \quad r_q\ ]^T$$

The overall set observations r are now related to the coefficients in x by the equation:

$$\mathbf{r} = C\mathbf{x} \tag{7.14}$$

where the matrix C is given by:

$$C = \begin{bmatrix} 1 & \cos(\theta_1) & \sin(\theta_1) & \cdots & \cos(q\theta_1) & \sin(q\theta_1) \\ 1 & \cos(\theta_2) & \sin(\theta_2) & \cdots & \cos(q\theta_2) & \sin(q\theta_2) \\ & & & \vdots & & \\ 1 & \cos(\theta_m) & \sin(\theta_m) & \cdots & \cos(q\theta_m) & \sin(q\theta_m) \end{bmatrix}$$

Solvability of the equation (7.14) depends on rank of the matrix C, yielding a straightforward solution characterization. More interesting is the fact that we may implement the LLSE solution to this equation recursively, due to the linear form of the observation equation (7.13) for a fixed set of directions. Such an approach obviates the need for inversion of the matrix C. In a higher dimensional case with many radius samples, this matrix may be quite large. Preliminary experience with such algorithms appear promising.

We can further imagine letting the descriptors $\mathbf{x}$ of the shape evolve according to a dynamical equation of the form:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t$$

Such an evolution would yield a corresponding dynamic object described by $r(\theta, t)$. With the addition of an observation equation, a linear state estimation problem is obtained:

$$\begin{aligned} \mathbf{x}_{t+1} &= A\mathbf{x}_t \\ r(\theta, t) &= c(\theta)\mathbf{x}_t \end{aligned}$$

where $c(\theta)$ is given by:

$$c(\theta) = [\ 1 \quad \cos(\theta) \quad \sin(\theta) \quad \cdots \quad \cos(q\theta) \quad \sin(q\theta)\ ]$$

We can imagine using such a formulation to estimate the shape of an evolving *non-convex* object over time. We leave further development for future work.

Chapter 8

Summary and Conclusions

In this thesis, we have contributed to the area of reconstruction from projections by our unification and extension of existing results, our concise characterization of the solutions to certain algebraic and geometric problems, and development of constrained reconstruction methodologies. Two specific physical problems were examined in detail. The projection and reconstruction of smooth, convex surfaces from curvature information, and the mathematically related problem of estimating ellipsoids from projections. In examining these problems, a common algebraic framework involving PSD symmetric matrices was identified, and led us to an extensive examination of this mapping and its properties in the space of symmetric matrices. Constrained solution of the algebraic problem of estimating a PSD matrix led to corresponding solutions to the physical problems. The incorporation of a dynamic element into this framework was straightforward, yielding a dynamic shape estimation problem. The tie between 1-dimensional shadow projections and support hyperplane observations led to a consideration of the consistency and smoothness of support-based reconstructions for use in constrained support-based reconstructions. Such reconstructions were demonstrated for the case of an ellipse.

In Chapter 2, projection mapping between symmetric matrices which captured the common algebraic framework of the curvature and ellipsoid problems was defined. An isometric isomorphism between this matrix projection mapping and an equivalent standard linear estimation form was provided. This isomorphism was based on a mapping of the problem to the space of symmetric matrices. This approach is simpler and more intuitive than many currently in use to solve such problems. Since

the symmetric matrices involved in the underlying physical problems of curvature and ellipsoid projection were required to be positive semi-definite, we next investigated the geometry of the set of PSD matrices. Various new results concerning the structure of this set were given. In particular, a representation of the PSD set in terms of a minimal set of its support halfspaces was provided. The normals to these halfspaces corresponded to rank-1 symmetric matrices. These insights were used to provide simple polyhedral approximations to the PSD set. Because the approximations were polyhedral, corresponding to a set of linear inequality constraints, they were convenient for inclusion in constrained optimization. In particular, we showed how tessellations of the unit sphere induced both interior and exterior polyhedral approximations to the PSD cone. A variety of other approximations were provided, including a procedure of obtaining a family of sets between support-based exterior and interior approximating sets. All these approximations shared the simple linear constraint property.

In Chapter 3 the inverse problem of estimating a symmetric matrix from a series of its projection mappings was considered. The isometric isomorphism of Chapter 2 was used to provide a complete solution to this problem without a PSD constraint. This unconstrained solution will remain valid for small enough perturbations in a consistent set of observations. This notion was made precise and an explicit condition provided under which a constraint is not needed. As a PSD constraint is required in the general case, constrained reconstructions were examined next. The polyhedral approximations to the PSD set of Chapter 2 were used to approximate PSD-constrained reconstructions. A novel iterative algorithm was then provided based on the rank-1 support halfspace characterization of the PSD set of Chapter 2. This algorithm, which is guaranteed to converge, successively approximates the boundary of the cone at the solution. In addition to this algorithm, other gradient-based iterative algorithms solving the problem were provided. The PSD-constrained matrix reconstruction algorithm were then extended to the case of arbitrary matrix interval constraints of the form $\underline{X} \leq X \leq \overline{X}$. Numerical experiments were provided demonstrating the various reconstruction schemes. The results of both this chapter and Chapter 2 have general usefulness outside of the narrow considerations of the mapping defined in (2.3). The techniques developed there may be applied to the broad class of linear problems involving symmetric matrix.

In Chapter 4 the first of our two physical problems was treated. The relationship between the n-dimensional surfaces of smooth, strictly convex objects and the m-

dimensional surfaces of their orthogonal projections, or shadows, was examined. The main results were local in nature, relating the curvature of a projection at a point to that of the generating surface. In particular, the relationship between the symmetric curvature Hessians of the projection at a point and the surface at the point's preimage was shown to be precisely given by the projection mapping defined in Chapter 2. This enabled us to use the results of Chapter 3 to give a complete characterization of the inverse problem of determining surface curvature from a series of projections. The required PSD constraint on the reconstructed Hessian was also easily handled using the constrained algorithms developed there. These local results were then related to global issues arising from the reconstruction of an entire surface from a set of projections based on curvature information.

Chapter 5 examined the problem of ellipsoid projection and reconstruction. We represented an ellipsoid by the symmetric PSD matrix of its quadratic form. The relationship of an ellipsoid to its projections induced a relationship between these corresponding symmetric matrices that was precisely captured by the matrix mapping of Chapter 2. This quadratic mapping between symmetric matrices was also the same one relating the curvature Hessian and its projections in Chapter 4. The results of Chapter 3 again provided a complete solution characterization to the problem of reconstructing an ellipsoid from a series of consistent projections. A solution for the case of noisy inconsistent projections was provided by the PSD-constrained algorithms of Chapter 3. In addition, the interval matrix constrained algorithms of that chapter provided the ability to impose natural geometric constraints on the reconstructed ellipsoid in terms of inner and outer shape bounds. A dynamic equation on the ellipsoid matrices was then incorporated to yield a class of estimation problems involving a dynamically evolving ellipsoid. We demonstrated how proper choice of the dynamical matrices allowed direct access to geometric properties of dynamic ellipsoid evolution. The special case of 1-dimensional projections, resulting in observations of the squared support function, was emphasized. In particular, the constrained reconstruction of both static and dynamic ellipses from a series of noisy support measurements was demonstrated through a series of numerical experiments. Here we showed the potential value of prior support consistency in the reconstruction of ellipsoids.

Chapter 6 continued this support oriented theme through an examination of issues arising in support-based reconstructions. In particular, the consistency of support sample sets and the smoothness of the corresponding reconstructions were explored.

Such support measurements arise from 1-dimensional projections of objects, as mentioned above. A classical result due to Rademacher which characterized support functions in the plane was extended to a general dimensional discrete setting. This extended result was in turn used to develop a local consistency test for a discrete set of support measurements that was valid for arbitrary dimensions and sample orientations. This result generalized the planar, uniform-angle consistency results of [8]. We went on to use our insights from this endeavor to develop and analyze support based discrete notions of curvature for both the 2-dimensional and 3-dimensional cases. Finally, we proposed global measures of smoothness based on the classical isoperimetric inequality. The use of this measure in a smoothness-weighted support reconstruction was demonstrated through a numerical experiment.

The results of thesis have contributed to the growing area of geometric, model-based reconstruction from projections, aiding the effort to broaden its applicability to higher dimensions and more general situations. Our investigation of the fundamental underlying algebraic structure of some of these problems has enabled this extension. Our formulations of these results are themselves straightforward and general. The potential use of these results for the 3-dimensional case in particular seems potentially useful and interesting.

Bibliography

[1] F. C. Crow. Shadow algorithms for computer graphics. *Computer Graphics*, 11(2):242–248, Summer 1977.

[2] L. Williams. Casting curved shadows on curved surfaces. *Computer Graphics*, 12(3):270–274, August 1978.

[3] D. F. Rogers and R. A. Earnshaw, editors. *Techniques for Computer Graphics.* Springer-Verlag, New York, 1987. pp. 152.

[4] N. Magnenat-Thalmann and D. Thalmann, editors. *Computer-Generated Images: The State of the Art.* Springer-Verlag, Tokyo, 1985. pp. 45, 356.

[5] D. Dudgeon. Extraction and recognition of silhouettes from range imagery. M.I.T. Stochastic Systems Group Seminar, November 13, 1987.

[6] P. L. Van Hove. Model-based silhouette recognition. M.I.T. Digital Signal Processing Group Seminar, November 18, 1987.

[7] P. L. Van Hove. *Silhouette-Slice Theorems.* PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, September 1986.

[8] J. L. Prince. *Geometric Model-based Etimation from Projections.* PhD thesis, Massachusetts Institute of Technology, January 1988.

[9] D. J. Rossi. *Reconstruction from Projections Based on Detection and Estimation of Objects.* PhD thesis, Massachusetts Institute of Technology, August 1982.

[10] J. G. Verly, P. L. Van Hove, R. I. Walton, and D. E. Dudgeon. Silhouette understanding system for laser-radar range imagery. In *Proceedings of the Meeting of the IRIS Specialty Group on Active Systems*, Monterey, CA, November 5–7, 1985.

[11] J. G. Verly, P. L. Van Hove, R. I. Walton, and D. E. Dudgeon. Silhouette understanding system. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Tokyo, Japan, April 7–11, 1986.

[12] A. S. Lele. Convex set reconstruction from support line measurements and its application to laser radar data. Master's thesis, Massachusetts Institute of Technology, April 1990.

[13] B. K. P. Horn and E. J. Weldon, Jr. Filtering closed curves. *IEEE Journal of Pattern Analysis and Machine Intelligence*, PAMI–8:665–668, 1986.

[14] B. L. Yen and T. S. Huang. Determining 3-d motion and structure of a rigid body using the spherical projection. *Computer Vision, Graphics, and Image Processing*, 21:21–32, 1983.

[15] S. A. Shafer. *Shadows and Silhouettes in Computer Vision.* Kluwer, Boston, 1985.

[16] W. N. Martin and J. K. Aggarwal. Occluding contours in dynamic scenes. In *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing*, pages 189–192, 1981.

[17] Y. C. Kim and J. K. Aggarwal. Rectangular parallelepiped coding: A volumetric representation of three-dimensional objects. *IEEE Journal of Robotics and Automation*, RA-2:127–134, 1986.

[18] Y. Bresler, J. A. Fessler, and A. Macovski. A bayseian approach to reconstruction from incomplerer projections of a multiple object 3D domain. *IEEE Journal of Pattern Analysis and Machine Intelligence*, PAMI–8(8):840–858, 1989.

[19] P. Van Hove and J. Verly. A silhouette slice theorem for opaque 3-D objects. In *International Conference on Acoustic and Speech Signal Processing*, pages 933–936, March 1985.

[20] B. K. P. Horn. Extended gaussian images. *Proceedings of the IEEE*, 72:1671–1686, December 1984.

[21] B. K. P. Horn. *Robot Vision.* MIT Press, Cambridge, 1986.

[22] M. Spivak. *A Comprehensive Introduction to Differential Geometry*, volume V. Publish or Perish, Boston, MA, 1975.

[23] C. C. Hsiung. *A First Course in Differential Geometry.* John Wiley & Sons, New York, 1981.

[24] S. S. Chern, editor. *Studies in Global Geometry and Analysis*, volume 4 of *Studies in Mathematics.* Prentice-Hall, 1967.

[25] D. J. Rossi and A. S. Willsky. Reconstruction from projections based on detection and estimation of objects-parts I and II: Performance analysis and robustness analysis. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, ASSP-32(4):886–906, 1984.

[26] Y. Bresler and A Macovski. 3-D Reconstruction from projections based with incomplete and noisy data by object estimation. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, ASSP–35:1139–1152, 1987.

[27] T. Bonnesen and W. Fenchel. *Theory of Convex Bodies.* BCS Associates, Moscow, Idaho, 1987.

[28] E. H. Wood, J. H. Kinsey, B. K. Robb, B. K. Gilbert, L. D. Harris, and E. L. Ritman. Applications of high temporal resolution, computerized tomography to physiology and medicine. In G. T. Herman, editor, *Image Reconstruction from Projections*, chapter 6, pages 247–279. Springer-Verlag, Berlin, 1979.

[29] Fritz Keinert. Inversion of k-plane transforms and applications in computer tomography. *SIAM Review*, 31(2):273–289, June 1989.

[30] G. M. Constantine. On the invertibility of the discrete radon transform. *SIAM Journal on Discrete Mathematics*, 2(3):300–306, August 1989.

[31] G. Strang. Inverse problems and derivatives of determinants. Center for Intelligent Control Systems Technical Report CICS-P-234, Massachusetts Institute of Technology, MIT, Room 35-311, Cambridge, MA, June 1990.

[32] W. J. Vetter. Vector structures and solutions of linear matrix equations. *Linear Algebra and its Applications*, 10:181–188, 1975.

[33] J. R. Magnus and H. Neudecker. The elimination matrix: Some lemms and applications. *SIAM Journal on Algebraic and Discrete Methods*, 1(4):422–449, 1980.

[34] F. J. H. Don. On the symmetric solutions of a linear matrix equation. *Linear Algebra and its Applications*, 93:1–6, 1987.

[35] A. Graham. *Kronecker Products and Matrix Calculus: with Applications.* Ellis Horwood Series in Mathematics and its Applications. Halsted Press, Chichester, 1981.

[36] D. Hua. On the symmetric solutions of linear matrix equations. *Linear Algebra and its Applications*, 131:1–7, 1990.

[37] P. Lancaster and M. Tismenetsky. *The Theory of Matrices.* Academic, Orlando, 1985.

[38] R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, 1987.

[39] R. D. Hill. On the cone of positive semidefinite matrices. *Linear Algebra and its Applications*, 90:81–88, 1987.

[40] G. P. Barker. Theory of cones. *Linear Algebra and its Applications*, 39:263–291, 1981.

[41] B. S. Tam. Generalized inverses of cone preserving maps. *Linear Algebra and its Applications*, 40:189–202, 1981.

[42] B. S. Tam. On the distinguished eigenvalues of a cone-preserving map. *Linear Algebra and its Applications*, 131:17–37, 1990.

[43] A. Berman and A. Ben-Israel. Linear equations over cones with interior: A solvability theorem with applications to matrix theory. *Linear Algebra and its Applications*, 7:139–149, 1973.

[44] P. Tarazaga. Eigenvalue estimates for symmetric matrices. *Linear Algebra and its Applications*, 135:171–179, 1990.

[45] A. Brøndsted. *An Introduction to Convex Polytopes.* Springer-Verlag, New York, 1983.

[46] H. Guggenheimer. *Applicable Geometry: Global and Local Convexity.* Applied Mathematics Series. Krieger, Huntington, New York, 1977.

[47] F. A. Valentine. *Convex Sets.* McGraw-Hill Book Company, New York, 1964.

[48] R. Fletcher. Semi-definite matrix constraints in optimization. *SIAM Journal on Control and Optimization*, 23(4):493–513, July 1985.

[49] J. Hilgertt, K. H. Hofmann, and J. D. Lawson. *Lie Groups, Convex Cones, and Semigroups.* Clarendon Press, Oxford, 1989.

[50] B. R. Barmish, M. Fu, and S. Saleh. Stability of a polytope of matrices: Counterexamples. *IEEE Transactions on Automatic Control*, 33(6):569–572, June 1988.

[51] Y. T. Juang, S. L. Tung, and T. C. Ho. Sufficient condition for asymptotic stability of discrete interval systems. *International Journal of Control*, 49(5):1799–1803, 1989.

[52] M. Mansour. Simplified sufficient conditions for the asymptotic stability of interval matrices. *International Journal of Control*, 50(1):443–444, 1989.

[53] Y. K. Foo and Y. C. Soh. Stability analysis of a family of matrices. *IEEE Transactions on Automatic Control*, 35(11):1257–1259, November 1990.

[54] C. B. Soh. Schur stability of convex combination of matrices. *Linear Algebra and its Applications*, 128:159–168, 1990.

[55] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems.* Prentice-Hall, Englewood Cliffs, 1974.

[56] D. C. Youla and H. Webb. Image restoration by the method of convex peojection: Part 1 – Theory. *IEEE Transactions on Medical Imaging*, MI-1:81–94, 1984.

[57] M. I. Sezan and H. Stark. Tomographic image reconstruction from incomplete view data by convex projections and direct fourier inversion. *IEEE Transactions on Medical Imaging*, MI-3:91–98, 1984.

[58] P. O. Farda and H. Stark. Tomographic image reconstruction using the theory of convex projections. *IEEE Transactions on Medical Imaging*, MI-7(1):45–58, 1988.

[59] K. C. Tam, V. Perez-Mendez, and B. MacDonald. 3-D object reconstruction in emission and transmission tomography with limited angular input. *IEEE Transactions on Nuclear Science*, NS-26(2):2797–2805, 1979.

[60] A. Gelb, editor. *Applied Optimal Estimation.* MIT Press, Cambridge, 1984.

[61] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, 1989.

[62] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, Cambridge, 1989.

[63] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation.* Prentice Hall, Englewood Cliffs, 1989.

[64] L. Cooper and D. Steinberg. *Introduction to Methods of Optimization.* Saunders, Philadelphia, 1970.

[65] G. R. Walsh. *Methods of Optimization.* John Wiley & Sons, London, 1975.

[66] E. Polak. *Computational Methods in Optimization; A Unified Approach.* Academic Press, New York, 1971.

[67] J. C. Allwright. Positive semidefinite matrices: Characterization via conical hulls and least-squares solution of a matrix equation. *SIAM Journal on Control and Optimization*, 26(3):537–556, May 1988.

[68] W. C. Karl and G. C. Verghese. Curvatures of surfaces and their shadows. *Linear Algebra and its Applications*, 130:231–255, 1990.

[69] J. A. Thorp. *Elementary Topics in Differential Geometry.* Springer, New York, 1979.

[70] M. E. Mortenson. *Geometric Modeling.* John Wiley & Sons, New York, 1985.

[71] M. P. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, New Jersey, 1976.

[72] Shankar Moni. A closed-form solution for the recovery of a convex polyhedron from its extended gaussian image. Machine Vision Laboratory, University of Hawaii. Personal correspondence.

[73] J. J. Koenderink. *Solid Shape.* MIT Press, Cambridge, MA, 1990.

[74] F. C. Schweppe. *Uncertain Dynamic Systems.* Prentice-Hall, New Jersey, 1973.

[75] J. A. K. Blokland, A. M. Vossepoel, A. R. Bakker, and E. K. J. Paulwels. Delineating elliptical objects with an application to cardiac scintigrams. *IEEE Transactions on Medical Imaging*, MI-6(1):57–66, 1987.

[76] S. Kaul,M.D., FACC, C. A. Boucher, M.D., FACC, J. B. Newell, B.A., D. A. Chesler, Sc.D., J. M. Greenberg,M.D., FACC, R. D. Okada, M.D., FACC, H. W. Strauss, M.D., FACC, R. E. Dinsmore, M.D., FACC, and G. M. Pohost, M.D., FACC. Determination of the quantitative thallium imaging variables that optimize detection of coronary artery disease. *Journal of the American College of Cardiology*, 7:527–537, March 1986.

[77] P. Lipson, A. L. Yuille, D. O'Keeffe, J. Cavanaugh, J. Taaffe, and D. Rosenthal. Deformable templates for feature extraction from medical images. Artifical Intellegence Laboratory, MIT. Personal correspondence, 1989.

[78] F. C. Schweppe and H. K. Knudsen. The theory of amorphous cloud trajectory prediction. *IEEE Transactions on Information Theory*, IT-14:415–427, 1968.

[79] O. E. Drummond, S. S. Blackman, and K. C. Hell. Multiple sensor tracking of clusters and extended objects. In *Proceedings of the 1988 Tri-Service Data Fusion Symposium*, Laurel, MD, May 19, 1988.

[80] S. Tsuji and F. Matsumoto. Detection of ellipses by a modified hough transform. *IEEE Transactions on Computers*, C-27(8):777–781, 1978.

[81] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–112, 1981.

[82] H. K. Yuen, J. Illingworth, and J. Kittler. Detection partially occluded ellipses using the hough transform. *Image and Vision Computing*, 7(1):31–37, February 1989.

[83] F. L. Bookstein. Fitting conic sections to scattered data. *Computer Graphipcs and Image Processing*, 9:56–71, 1979.

[84] T. Nagata, H. Tamura, and K. Ishibashi. Detection of an ellipse by use of a recursive least-squares estimator. *Journal of Robotic Systems*, 2(2):163–177, 1985.

[85] D. E. McClure and R. A. Vitale. Polygonal approximations of plane convex bodies. *Journal of Mathematical Analysis and Applications*, 51:326–358, 1975.

[86] C. T. Chen. *Introduction to Linear System Theory.* Holt, Rinehart, and Winston, New York, 1970.

[87] M. I. Shamos. *Computational Geometry.* PhD thesis, Yale University, December 1977.

[88] D. T. Lee and F. P. Preparata. Computational geometry — a survey. *IEEE Transactions on Computers*, C-33:1072–1101, 1984.

[89] J. P. Greschak. *Reconstructing Convex Sets.* PhD thesis, Massachusetts Institute of Technology, February 1985.

[90] R. R. Kim. *Matrix Algorithms for Bilinear Estimation Problems in Chemometrics.* PhD thesis, Massachusetts Institute of Technology, June 1985.

[91] J. M. Humel. Resolving bilinear data arrays. Master's thesis, Massachusetts Institute of Technology, 1986.

[92] H. Busemann. *Convex Surfaces.* Interscience Publishers, New York, 1958.

[93] A. V. Pogorelov. *Extrinsic Geometry of Convex Surfaces.* Translations of Mathematical Monographs. American Mathematical Society, Providence, 1973.

[94] A. V. Pogorelov. *The Minkowski Multidimensional Problem.* John Wiley & Sons, New York, 1978.

[95] L. A. Santaló. *Intergral Geometry and Geometric Probability.* Encyclopedia of Mathematics and its Applications. Addison-Wesley, London, 1976.

[96] H. Rademacher. Über eine funktionale ungleichung in der theorie der konvexen körper. *Mathematische Zeitschrift*, 13:18–27, 1922.

[97] J. Telgen. Minimal representation of convex polyhedral sets. *Journal of Optimization Theory and Applications*, 38(1):1–24, September 1982.

[98] J. C. G. Boot. On trivial and binding constraints in programming problems. *Management Science*, 8:419–441, 1962.

[99] J. Telgen. *Redundancy and Linear Programs.* Mathematisch Centrum, Amsterdam, 1981.

[100] M. C. Cheng. Generalized theorems for permanent basic and nonbasic variables. *Mathematical Programming*, 31:229–234, 1985.

[101] M. C. Cheng. General criteria for rendundant and nonredundant linear inequalities. *Journal of Optimization Theory and Applications*, 53(1):37–42, April 1986.

[102] H. C. P. Berbee, C. G. E. Boender, A. H. G. Rinnooy Kan, C. L. Scheffer, R. L. Smith, and J. Telgen. Hit-and-run algorithms for the identification of nonredundant linear inequalities. *Mathematical Programming*, 37:184–207, 1987.

[103] J. L. Prince. Estimation algorithms for reconstructing a convex set given noisy measurements of its support lines. LIDS Technical Report LIDS-P-1638, Massachusetts Institute of Technology, Cambridge, MA, 1987.

[104] H. S. M. Coxeter. *Introduction to Geometry.* John Wiley & Sons, New York, 1969.

[105] H. H. Johnson and J. Osaka. An isoperimetric inequality form polyhedra. *American Mathematical Monthly*, 81:58–61, 1974.

[106] E. A. Lord and C. B. Wilson. *The Mathematical Description of Shape and Form.* John Wiley & Sons, New York, 1984.

[107] R. Sauer. *Differenzengeometrie.* Springer, Berlin, 1970.

[108] G. E. Forsyth and W. R. Wasow. *Finite-Difference Methods for Partial Differential Equations.* John Wiley & Sons, New York, 1960.

[109] S. K. Godunov and V. S. Ryabenkii. *Difference Schemes.* North-Holland, Amsterdam, 1987.

[110] C. Bandle. Isoperimetric inequalities. In P. M. Gruber and J. M. Wills, editors, *Convexity and its Applications*, pages 30–48. Birkhäuser Verlag, Basel, 1983.

[111] H. Groemer. Stability properties of geometric inequilities. *American Mathematical Monthly*, 95(5):382–394, May 1990.

[112] G. Poole and T. Boullion. A survey on m-matrices. *SIAM Review*, 16(4):419–427, October 1974.

[113] J. W. Roach and J. K. Aggarwal. Computer tracking of objects moving in space. *IEEE Journal of Pattern Analysis and Machine Intelligence*, PAMI–1:127–135, 1979.

[114] J. A. Webb and J. K. Aggarwal. Visually interpreting the motion of objects in space. *IEEE Computer*, 14:40–46, 1981.

[115] S. Ullman. Interpretation of structure from motion. M.I.T. Artificial Intellegence Memo 476, Massachusetts Institute of Technology, Cambridge, MA, 1976.

[116] L. Ljung and T. Soderstrom. *Theory and Practice of Recursive Identification.* MIT Press, Cambridge, 1983.

[117] C. C. Lin and R. Chellappa. Classification of partial 2-D shapes using fourier descriptors. *IEEE Journal of Pattern Analysis and Machine Intelligence*, PAMI–9(5):686–690, 1987.

[118] C. W. Richard, Jr. and H. Hemami. Identification of three-dimensional objects using fourier descriptors of the boundary curve. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC–4(4):371–378, July 1974.

[119] C. T. Zahn and R. Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, C–21(3):269–281, March 1972.

[120] N. L. Max. Sphereical harmonic molecular surfaces. Lawrence Livermore National Laboratory. Preprint.

[121] R. B. Schudy and D. H. Ballard. A computer model for extracting moving heart surfaces from four-dimensional cardiac ultrasound data. In *Proceedings of the Sixth IEEE Conference on Computer Applications in Radiology and Computer-Aided Analysis of Radiological Images*, pages 366–376, Newport Beach, 1979. IEEE.

[122] A. Jouan. Analysis of sequences of cardiac countours by fourier descriptors for plane closed curves. *IEEE Transactions on Medical Imaging*, MI-6:176–180, 1987.

[123] R. Tello, R. W. Mann, and D. Rowell. Fourier descriptors for biomedical computer graphics. In *Proceedings of the Eighth Annual Conference of the Engineering in Medicine and Biology Society*, pages 1062–1064, 1986.