Syntactic and Semantic Image Representations for Computer Vision

by

Bradley Joseph Horowitz

B.S., COMPUTER SCIENCE
UNIVERSITY OF MICHIGAN, 1988

SUBMITTED TO THE MEDIA ARTS AND SCIENCES SECTION, SCHOOL OF ARCHITECTURE AND PLANNING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1991

Author ....
May 10, 1991

Certified by ....
Alex P. Pentland
Associate Professor
Computer Information and Design Technology
Thesis Supervisor

Accepted by ....
Stephen Benton
Chairperson
Departmental Committee on Graduate Studies

Syntactic and Semantic Image Representations for Computer Vision

by

Bradley Joseph Horowitz

Submitted to the Media Arts and Sciences
Section, School of Architecture and Planning
on May 10, 1991, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Computer vision requires the processing of images at various levels of abstraction. This thesis explores two image representations for vision which can be classified as "syntactic and "semantic respectively. As an exploration into syntactic (signal level) representations, an image coding technique based on the statistical relationship between subbands of the wavelet transform is explored. A sample implementation is described, which shows how this technique can be integrated into standard vector quantization coding schemes. A formulation for the recovery of rigid and non-rigid motion from optical flow is presented as an exploration of semantic (content based) image compression. This framework is based on the dynamic behavior of deformable models, and allows for closed form solution. This technique has applications for image understanding and the interpretation of visual motion.

Thesis Supervisor: Alex P. Pentland
Title: Associate Professor
Computer Information and Design Technology

Acknowledgments

Dad, for teaching me how to work.
Mom, for her love and support.
Gramma and Grampa, for their well timed visits.
Ramesh Jain, for his wisdom and friendship.
Terry Weymouth, for getting me started.
Laureen and Bea, for putting up with me.
Linda Peterson, for caring when I didn't.
Martin Friedmann, for being Martin.
Bobby and Gman, for being there and letting me be there sometimes too.
Stan Sclaroff, for his giving heart.
Eero Simoncelli, for letting me play around with EPIC
and consistently good advice.
The THINGWORLD programming posse.
Mary, for being my friend.
Nicholas, for creating such a wonderful place to learn.
VISMOD, for somehow striking the balance between being a
high-performance research engine, and a great place to hang out.
Tejasi, for her pure heart and clear head.
Gurumayi's Boston Ashram, for its boundless generosity and support.

and lastly Sandy, for his patience, encouragement and generosity.

Dedication

This thesis is dedicated to Gurumayi Chidvilasananda.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

Early in the development of the field of vision, a natural distinction between "low-level" and "high-level" visual processes was discovered. Biologically, these processes can be classified by *where* they occur in the visual system. The lowest level processing occurs in the architecture of the retina itself. As one traces the visual system back to the cortex, these low-level mechanisms are combined in increasingly complex ways which yield high-level functionality.

Image coding and machine vision have a similar relationship. Image coding deals with finding a representation that can be expressed in the minimum amount of information. Coding can be considered a *syntactic* representation in that it looks for image-level relationships (correlations) in the image signal itself. Machine vision also looks for correlations; rather than 2-D image correlations, it seeks the correlations that are meaningful for robotics and other applications. By discovering such meaningful correlations machine vision seeks to transform the image signal into a *semantic* representation.

The representations of machine vision and image coding form a continuum. This thesis consists of two individual research endeavors which are at opposite ends of this continuum, but both aimed at the problem of representing images and their content. The first part of the thesis, Chapters 2 through 5, explores a technique for image compression which is based on the 2-D image-level relationship between subbands of the wavelet transform. The second part, Chapters 6 through 10, addresses the high-level representation of 3-D motion. The final chapter briefly hints at how these two techniques might be combined to form a unified image interpretation. I will describe in detail two specific research projects which are grounded in these different ways of treating and managing images.

1.2 Collaborations and Supporting Research

It should be mentioned that much of the work presented here is of a collaborative nature. The algorithm for the interpretation of non-rigid motion was developed with my thesis advisor Alex Pentland. The platform on which it is implemented, ThingWorld, is the result of continued effort by a number of people: Alex Pentland, Stanley Sclaroff, Martin Friedmann, Irfan Essa and Thad Starner. Related work by Stanley Sclaroff shows how similar techniques can be used to fit range data and recognize objects, and the reader is encouraged to also consult that work for more information. The initial experiments on vector quantization were inspired by the work of Takenori Seno and Bernd Girod. Ted Adelson's and Eero Simoncelli's EPIC (Efficient Pyramid Image Coder) formed the platform for investigations in image coding. When the pronoun "we" is used in the text, it refers to Alex Pentland and myself. I am thankful to all of these colleagues and apologize for any omissions.

It should also be noted that much of the material presented here has been published in journals and conference proceedings. The work on non-rigid motion is derived from [22] and the results in fractal image coding were first presented in [26].

Syntactic Image Representations:

A Practical Approach to Fractal Image Coding

Chapter 2

Fractal Image Coding

Fractal techniques for image compression have recently attracted a great deal of attention. Unfortunately, little in the way of practical algorithms or techniques have been published. We present a technique for image compression that is based on a very simple type of iterative fractal. In our algorithm a wavelet transform (quadrature mirror filter pyramid) is used to decompose an image into bands containing information from different scales (spatial frequencies) and orientations. The conditional probabilities between these different scale bands are then determined, and used as the basis for a predictive coder.

We find that the wavelet transform's various scale and orientation bands have a great deal of redundant, self-similar structure. This redundant structure is, however, in the form of multi-modal conditional probabilities, so that linear predictors perform poorly. Our algorithm uses a simple histogram method to determine the multi-modal conditional probabilities between scales. The resulting predictive coder is easily integrated into existing subband coding schemes. Comparison of this fractal-based scheme with our standard wavelet vector coder on 256 x 256 grey-level imagery shows up to a two-fold gain in coding efficiency with no loss in image quality, and up to a four-fold gain with small loss in image quality. Coding and decoding are implemented by small table lookups, making real-time application feasible.

2.1 Introduction

Fractal geometry was introduced by Mandelbrot [17] in the late seventies, and has had important consequences in a number of domains. In image processing and generation fractals have been important because they can describe and generate natural-looking images with (literally) infinite detail using only a small number of simple rules and parameters. The simplicity of these rules lead the computer graphics community to immediately adopt fractals as their primary stochastic modeling tool.

In the image processing and compression community, fractals have generated a great deal of interest because the prospect of solving the inverse problem, that of automatically recovering simple rules that describe complex imagery, is extremely attractive. Pentland [18] was perhaps the first to apply fractals to image processing, using estimates of fractal parameters to perform texture segmentation and shape extraction. More recently, Barnsley and Sloan [5] have proposed using iterated function systems (IFS) to achieve massive compression of some images, while Walach & Karnin have proposed a "yardstick" technique to achieve a fractal encoding of images.

Most common fractals exhibit a type of statistical self-similarity, that is, there is a fixed pattern of relationships that exists between the fractal's values at different scales of examination. Brownian fractals, for instance, exhibit a self-affine correlation structure. In consequence, we may characterize these fractal signals by discovering the pattern of relationships that exists within and between the different scales.

As an example, let us examine the appearance of some different-scale subbands of a typical image. Figure 2-1 shows a three-level Laplacian pyramid computed from an image, a decomposition of the signal into roughly octave-wide subbands. This transform was suggested by Burt and Adelson [7], and uses separable Gaussian linear filters to recursively bandsplit the image, generating a pyramid-like data structure that provides a compact, multi-scale representation of an image. Even a cursory examination of these subbands makes it clear that there is great similarity between the various scales. In large part this is due to the self-similarity of edges; a perfect edge appears the same at all scales and hence appears in the same location in each subband.

The image model that we have adopted is a generalization of the behavior of edges in subbands, and characterizes images as a simple type of recursively-defined, non-linear fractal.

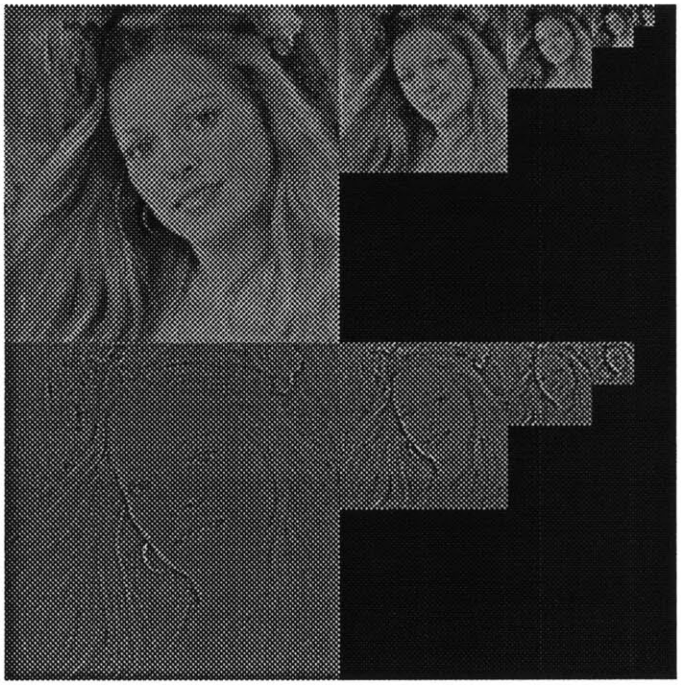

Figure 2-1: 3-level Laplacian pyramid of the image *face*

In particular, our model is that (1) all subbands of the same orientation consist of unions of the same primitive elements (e.g., edge fragments, textures), and that (2) for all subbands of the same orientation there is a fixed, one-to-many function that maps that subband's elements to the elements of the next-higher frequency subband. In the case of perfect edges the subband-to-subband mapping is simply the identity operation, however in most cases the mapping is not only one-to-many but also non-deterministic. Our model defines a simple type of iterative fractal, and when applied recursively can generate a wide class of imagery.

The main idea behind our approach to image compression is to characterize the set of mappings that exist between subbands, and to use this knowledge to achieve greater compression ratios. By use of a standard vector quantization approach we are able to characterize the patterns that exist within subbands, and by a simple histogramming approach we are able to statistically characterize the mapping between subbands. Using these characterizations we are then able to remove redundancies that exist between, and within, subbands.

Our work is similar in spirit to image extrapolation or interpolation techniques that attempt to predict high frequency detail (or subbands) from lower frequency content (or subbands). Extrapolation research, however, has focused almost exclusively on the use of linear and quasi-linear filters, whereas our approach is based on the observation that the conditional probabilities that relate different image subbands are inherently non-linear and multi-modal. Our work is also similar to the iterated function systems research of Barnsley [5]. In our approach, however, we limit ourselves to a much smaller class of iterated functions, thus simplifying the search for the correct set of iterative mappings.

2.2 A Testbed System

In this paper we will explain and demonstrate our fractal technique by applying it to a subband coding system using a wavelet decomposition (also referred to as a quadrature mirror filter (QMF) pyramid) and vector quantization (VQ). Several properties of the wavelet transform make it especially suitable for VQ. In particular, each oriented subband generated by the decomposition exhibits internal structure: the horizontal, vertical, and diagonal subbands show horizontal, vertical and diagonal structure respectively due to the frequency response of the filter. It is important to remember, however, that our technique may potentially be adapted to

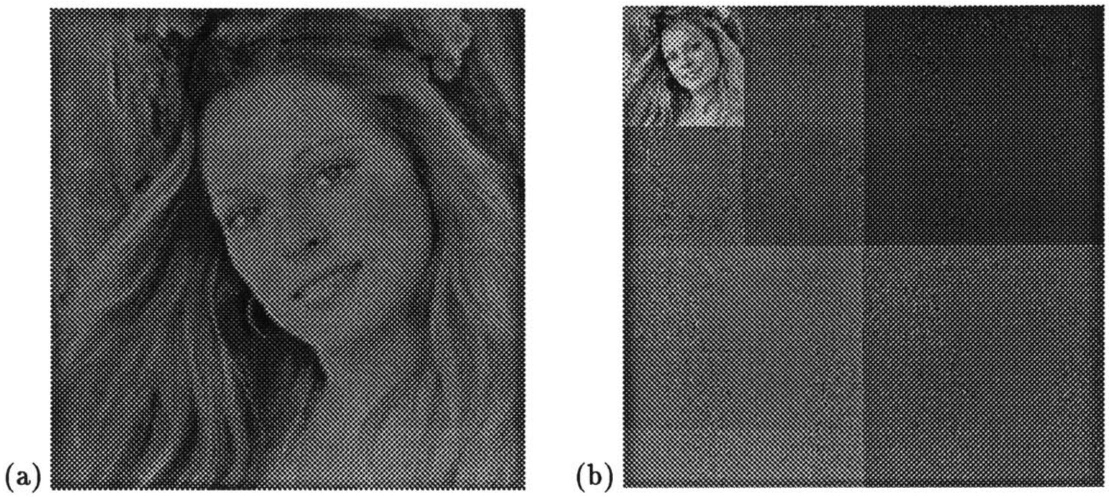

Figure 2-2: (a) Original *face* image and (b) its wavelet (QMF) decomposition

any subband coding system.

Our testbed system used in this paper is intended as a simple, general-purpose image compression facility rather than being optimized for low bit rates. In particular, because the system employed here is intended for near-real-time encoding/decoding, it avoids complex bit allocation schemes. Typical bit rates for the system used in this paper are 0.57 bits-per-pixel (bpp) for a 30 dB SNR (approximately 35 dB peak SNR) using 256 x 256 eight-bit grey-level imagery. The details of this testbed system are described in the following sections. The wavelet transform (QMF pyramid) employed is based on the nine-tap filter developed in our laboratory by Adelson and Simoncelli [1].

2.3 The Wavelet Transform

Perhaps the earliest study using the wavelet transform or QMF pyramid for image compression was by Adelson and Simoncelli [1], while Vetterli was the first to use quadrature mirror filters for image compression [33]. However other researchers, such as Mallat [16], Gharavi and Tabatabai [10], and Tran *et. al.* [31], also suggested use of the wavelet transform at about the same time.

The wavelet transform consists of a set of spatially-localized orthonormal linear filters which split an image into oriented spatial frequencies bands. An important property of the wavelet transform (QMF pyramid) is that can be constructed by recursive application of a base filter to successive lowpass subbands, requiring only $O(n)$ operations, where n is the number of image pixels.

The result of this transform is a set of subbands which are localized in scale (spatial frequency), orientation, and space. An example of such a transform is shown in Figure 2-2; for additional detail see Simoncelli and Adelson [28].

Chapter 3

Vector Quantization

3.1 Standard Vector Quantization

The QMF transform does not in itself reduce the amount of data needed to represent the signal. However, the entropy of the data is reduced, and therefore standard coding techniques can be applied in order to acheive compression. In this initial implementation, we have investigated using vector quantization as a coding technique. VQ is well suited to our application, since as well as efficiently coding blocks of coefficients, it also inherently performs pattern matching. Therefore, the same technique used for compression can also be used to determine the fractal statistics of the image.

Vector quantizers map an input signal onto a set of finite reproduction vectors, known as the *codebook.* In image compression, each input signal is a 2D k x l image patch. Once the codebook is constructed, pattern matching occurs between the input vector and the codebook entries. Ordering schemes, such as uniform lattices and K-d trees, have been suggested to prune search times in the coding and generation processes.

Our implementation uses the Equitz nearest neighbor (ENN) algorithm to approximate a minimum distortion quantizer over the set of training vectors. The algorithm iterates by replacing the two "nearest neighbor" vectors in the codebook by a vector which optimally encodes their constituency. Codebook distance is measured using mean squared error, and vectors are weighted by the number of samples which depend on them. Typically the ENN algorithm iterates over the entire set of training vectors until it converges to a solution; we have chosen to incrementally build the codebook and constrain its maximum size to 256 entries. This greatly reduces the combinatorics of codebook generation while only slightly affecting the quality of the resulting codebook. It should be stressed that the fractal technique is not tightly coupled to the this particular VQ framework.

3.2 Uniform Lattice Quantization

Standard VQ tends to be quite sensitive to its training set. In our experience, we found that the performance of Equitz VQ varied greatly, depending on content of the training images and subsequent test images. Moreover, for those images on which VQ did poorly, there was no way to incrementally increase the quality of the image. The fidelity of the reconstructed image was upper bounded by the content of the static, predetermined codebook.

These limitations of the standard VQ scheme are well-known, and much recent research has focused on addressing and overcoming them. Hierarchical VQ, edge-preserving VQ and adaptive VQ are but a few of the proposed improvements to the basic paradigm. Rather than employ one of these hybrid schemes, we sought an alternative method which would provide us with arbitrary reconstruction fidelity (with tradeoff in bitrate), and no dependence on a statistical training set.

3.2.1 EPIC

The EPIC system, developed in our lab by Adelson and Simoncelli [28], is a pyramid coder based on scalar quantization of QMF transform coefficients. We adapted this system to act as a uniform vector quantizer by grouping quantized coefficients in k x l blocks. Each k x l block is then assigned a codeword by simply concatenating the coefficients. (An efficient *shift* and *or* operation in implementation terms.) Thus the codeword for each block can be one of $(k * l)^n$ values, where n is the number of values that the quantized transform coefficients may assume. The number of codewords is thus exponential with respect to n. However it is important to stress that this is implicit VQ; there is no codebook as such, and images are coded and decoded without search. Moreover, even for large n the histogram of the coefficient values is strongly biased towards zero; therefore the entropy of actual codewords is, in practice, low.

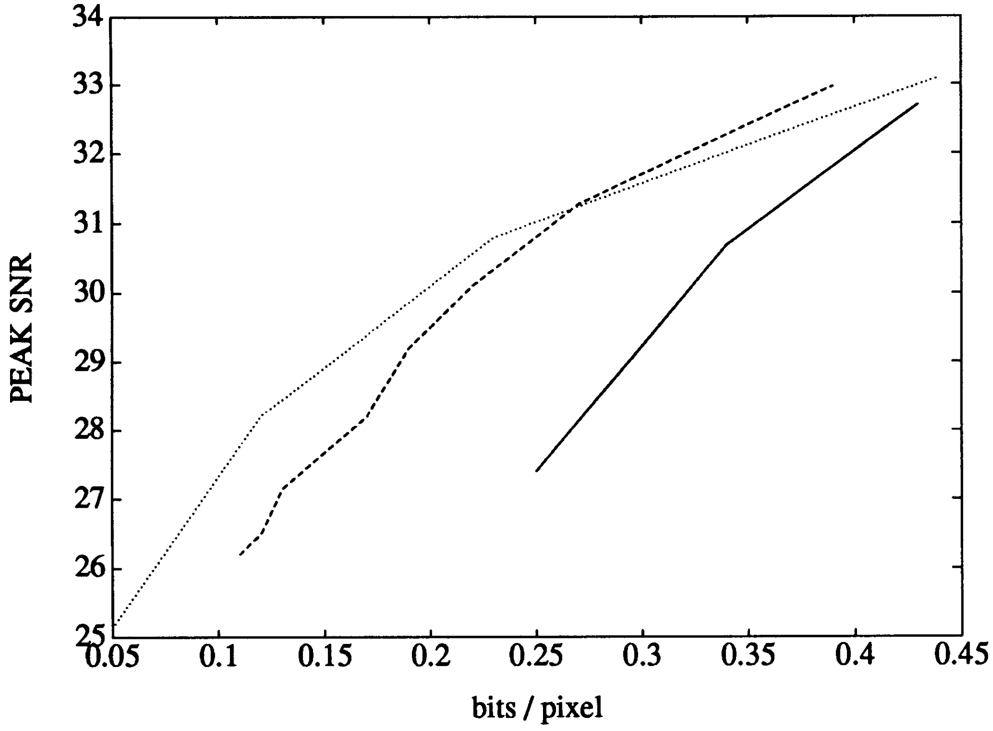

Figure 3-1: Comparison between (a) 4D uniform lattice quantizer (dotted line), (b) standard run-length and Huffman coding of QMF coefficients (dashed line), and (c) JPEG (solid line). The results for the ENN VQ of Section 3.1 are of higher quality and greater bitrate, and hence lie to the top right of the graph.

3.2.2 Results of Uniform Lattice Quantizer

Figure 3-1 shows a comparison between the four dimensional lattice quantizer described in Section 3.2, a scalar quantizer with run-length and huffman coding of coefficients and JPEG. The JPEG results reflect a simple doubling of the coefficient tolerances in the standard JPEG implementation, with no attempt to optimize performance for the specific image tested. The figure shows that the Lattice Quantizer is an efficient method at the extremely low bitrates we have been investigating. Figures 3-2 a shows several examples of extremely low bitrate coding using the lattice quantizer.

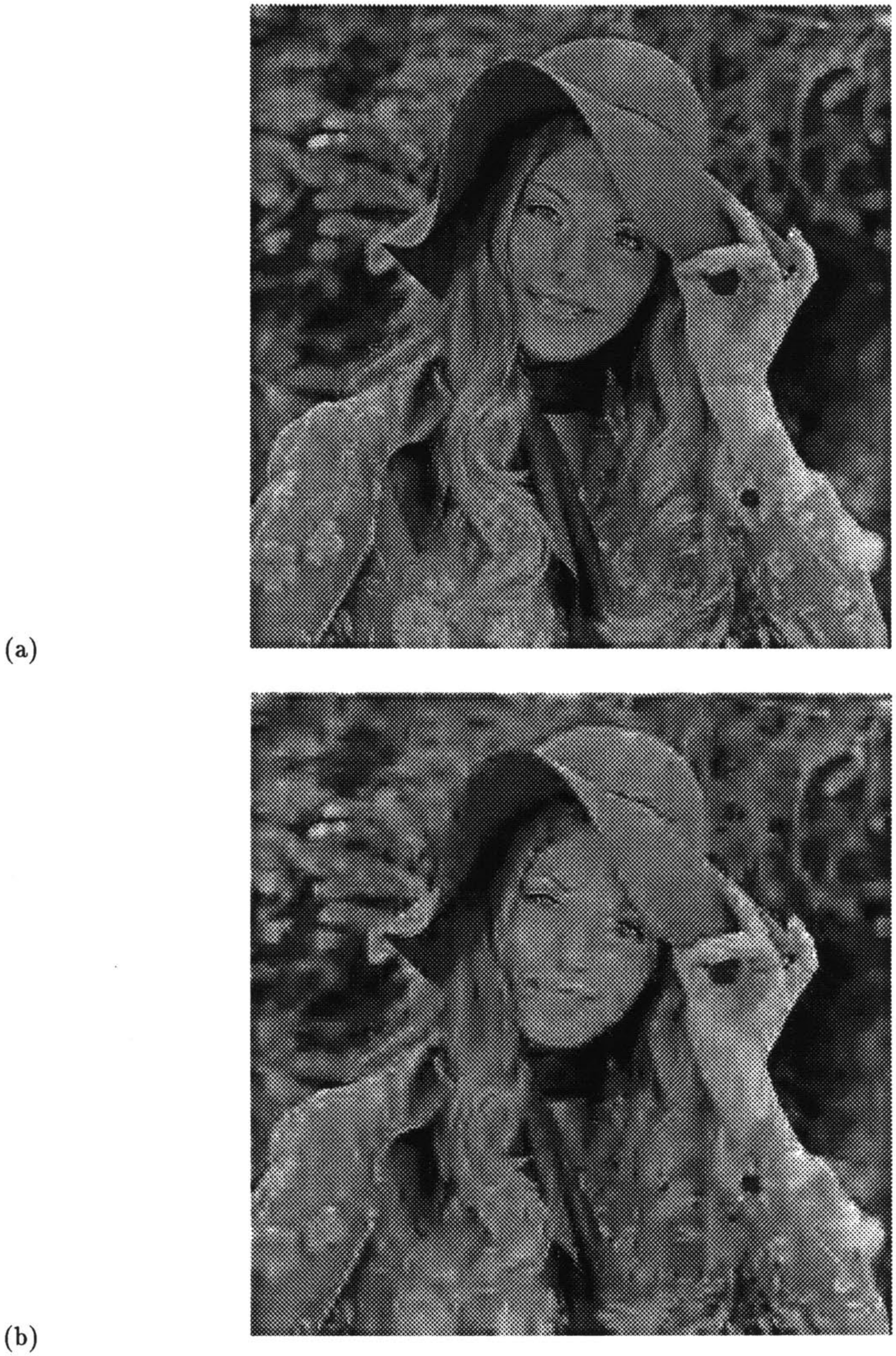

Figure 3-2: Images coded using the lattice quantizer described in Section3.2. (a) .19 bpp at 29.76 dB peak SNR (b) .096 bpp at 27.14 dB peak SNR.

Chapter 4

Determining and Using Relationships Between Subbands

4.1 Fractal Coding

The output of the VQ stage is a set of codes for each of the horizontal, vertical and diagonal subbands within each of the scales (pyramid levels) included in the wavelet transform. The codes for a lower frequency subband will contain one-fourth the number of entries contained by the next-higher frequency subband. The goal of our fractal-based encoding scheme is to first estimate the conditional probabilities between various subband's codes, and then use these statistical relationships in a predictive coding algorithm.

In the pyramid subband structure shown in Figure 2-2(b), an k x l patch in one subband corresponds spatially to four k x l patches in the next-higher frequency subband. Similarly, a single entry in that subband's coded representation corresponds to four entries in the coded representation of the next-higher frequency subband. Determining the conditional probabilities between these codes is thus quite simple. As images are coded, the mappings between the VQ codes for each subband and the next-higher frequency subband are tallied to obtain a histogram of the frequency of each of the mappings. Separate histograms are required for each of the four lower-to-higher frequency mappings. These histograms are called the *prediction lookup tables.*

Once this frequency histogram is constructed, the conditional probabilities between the various subband codings can be analyzed to determine an optimal lossless (with respect to the

VQ coding) encoding scheme. In this scheme each VQ-coded entry of each band is re-coded by comparing it to the codes most frequently associated with the VQ code at the corresponding location in the next-lower frequency band. Typically the higher frequency band's VQ code is one of the $2^n - 1$ codes found most frequently in the prediction lookup table associated with the lower frequency band's VQ code. The VQ code for the higher frequency band can then be re-coded by use of an n bit index into that prediction lookup table.

Thus each VQ code in the higher frequency band is re-coded by the following token:

n prediction index bits	m bit VQ index (for prediction failures)

The prediction index indicates which of the $2^n - 1$ most frequent entries in the prediction lookup table is the correct lower-to-higher frequency mapping. In the event that the correct code for the higher frequency subband is not one of the $2^n - 1$ most frequent predictions, the 2^nth index is reserved to indicate that the next token is a standard VQ codebook index. An optimal value for n is determined by minimizing

$$\frac{bits}{token} = P_n n + (1 - P_n)(n + m) = n + (1 - P_n)m \tag{4.1}$$

where n = number of bits used to index into the prediction lookup table, P_n is the percentage of low-to-higher frequency mappings accounted for by the most frequent $2^n - 1$ mappings, and m is the number of bits used to encode a full codebook index in case of a prediction failure. The optimal n is determined by analyzing the prediction lookup tables.

In our system the prediction index tokens and the VQ tokens resulting from the above recoding are separately passed through standard Huffman and run-length encoders, thus forming the code that is finally transmitted. The length of these final codes, rather than entropy estimates, are the basis for the all of bit rates quoted in this paper.

4.2 Results of Fractal Coding Technique

Table 4.1 shows how the percentage of re-codings, and the resulting bit rate, varies as a function of the number of bits n used to encode the lower-to-higher frequency mapping. These statistics are calculated using the testbed wavelet-and-VQ system described in the previous section, and

Table 3: Bit Rate as a Function of Predictions over Training Set					
		Percentage Encoded			
Predictions	Bits	Horizontal	Vertical	Diagonal	Bit Rate
0	0	0.0	0.0	0.0	.57
1	1	33.96	34.65	28.61	.47
3	2	57.75	61.40	52.91	.43
7	3	78.98	82.01	75.13	.36
15	4	93.53	96.06	91.81	.35
31	5	99.88	99.98	98.94	.38

are averages over a set of ten standard 256 x 256 eight-bit grey-scale images. A 3-level wavelet transform (QMF pyramid) was used, and the vector quantizer used a vector size of 4 x 4 pixels. Separate codebooks were used for each orientation within the pyramid; all coding has occurred at the fixed rate of 8 bits per vector in order to simplify and facilitate determining the statistical relationship between subbands.

As Table 4.1 shows, our fractal coding scheme produces up to a mean improvement of 1.5 times the testbed's original 0.57 bpp average coding rate. Because the fractal scheme is lossless with respect to the original VQ coding, the mean 30 dB SNR (35 dB peak SNR) is maintained even at a mean bit rate of 0.35 bpp. It is important to remember that our fractal coding scheme can potentially be combined with any subband coder.

Figure 4-1(a) shows the original *girl* image, at 256 x 256 pixel and eight-bit grey-level resolution. This image is coded by the testbed system at 0.57 bpp with 28.9 SNR (34.9 dB peak SNR). By combining our fractal coding scheme (using five prediction index bits) with the testbed coding system a typical bit rate of 0.35 bpp is achieved. Since the technique is lossless with respect to the original VQ coding, the image shows no additional degradation as a function of bit rate. Figure 4-1(b) shows the reconstructed image at 0.31 bpp with 28.9 dB SNR (34.9 dB peak SNR).

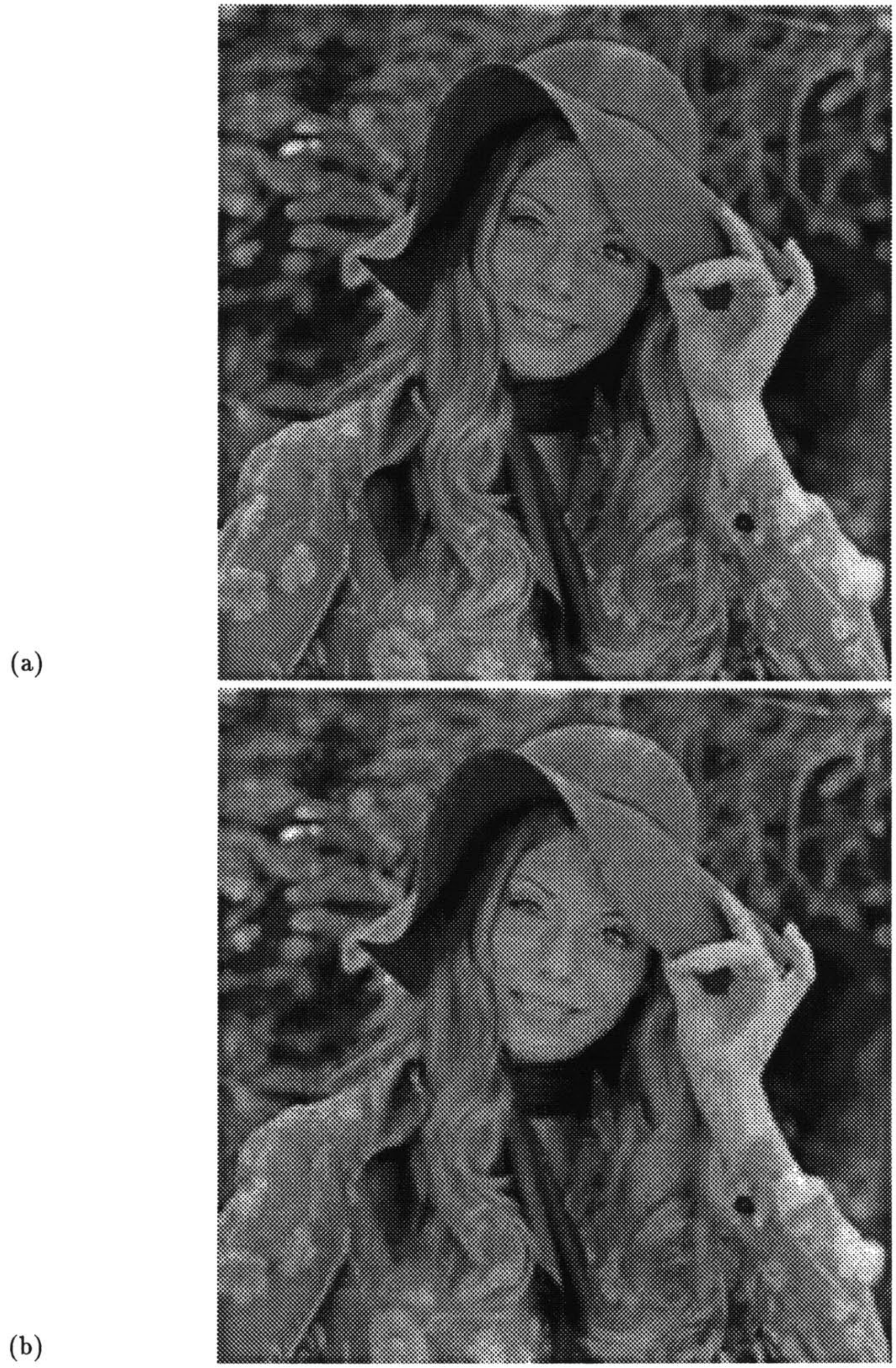

Figure 4-1: (a) Original and (b) reconstructed *girl* image, 0.31 bpp at 28.9 dB SNR (34.9 dB peak SNR) using our fractal coding technique.

4.3 Lossy Image Coding

The scheme described above gives lossless compression with respect to the original VQ coding. The same approach can be readily adapted to lossy coding, by simply removing the "catch" entries. That is, rather than reserving the 2^nth entry of the prediction index as an indicator that a standard VQ codebook entry follows, we can instead transmit the prediction codebook index that is closest to the correct VQ entry. In our experiments, using 10 standard 256 x 256 eight-bit grey-scale imagery, this approach has resulted in a three-fold mean increase in coding efficiency as compared to the original testbed coding system. Typical results are bit rates of approximately 0.125 bpp with approximately 28 dB SNR (approximately 33 dB peak SNR).

An example of this type of fractal coding is shown in Figure 4-2. Figure 4-2(a) shows the original image, and Figure 4-2(b) shows the lossy fractal coding at 0.125 bpp with 26.8 dB SNR (31.9 dB peak SNR).

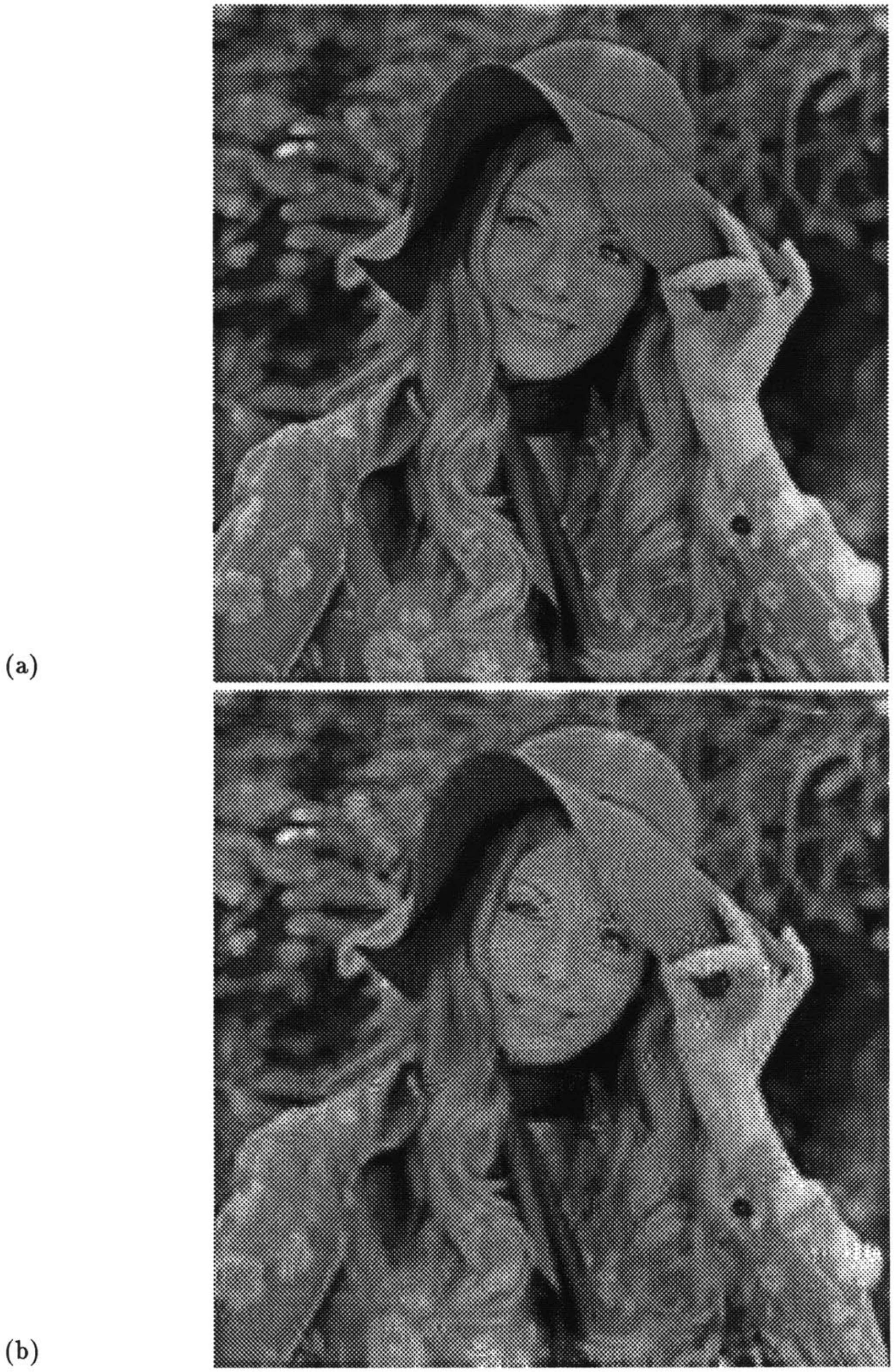

Figure 4-2: (a) Original and (b) reconstructed *girl* image, 0.125 bpp at 26.8 dB SNR (31.9 dB peak SNR) coded using the lossy fractal technique described in Section 4.3.

Chapter 5

Summary

5.1 Conclusion

We have described a fractal coding technique based on a very simple type of iterative fractal. In our algorithm a wavelet transform (QMF pyramid) is used to decompose an image into bands containing information from different scales (spatial frequencies) and orientations. The conditional probabilities between these different scale bands are then determined, and used as the basis for a predictive coding scheme.

We find that the wavelet transform's various scale and orientation bands have a great deal of redundant, self-similar structure. This observation lends support to the assertion that most images conform to our fractal model of image structure. The resulting predictive coder is easily integrated into existing subband coding schemes, and produces an average 1.5-fold gain in coding efficiency with no loss in image quality, and up to a four-fold gain with slight loss in image quality. Coding and decoding are implemented by small table lookups, making the scheme practical for real-time applications.

5.2 Epilogue: Motion Coding

Many of the same techniques which are described here for still images can be used to code image sequences as well. Figure 5-1 shows one second of video coding at 64Kbit / sec. A simple differencing method has been used to exploit interframe redundancy. Image differencing is the simplest way of coding image sequences. Since motion between frames is likely to be small, pixel brightness tends to vary slowly. Differencing fails when blocks of pixels move globally, either due to movement of an object or of the camera (panning.) For instance if the entire image is translated by five pixels, the sparsity of the difference image is lost. However, it is clear that if this global motion could be extracted, reconstruction would be trivial.

The techniques described in this section can be extended to serve as input to the system described in the next section. Three-dimensional QMF filters can be applied to extract local measurements of image motion as described in [12]. The second half of this thesis addresses the problem of how to convert these local, noisy 2-D measurements into a coherent 3-D representation. The output of such a system could then be used in a sophisticated coding scheme, although this is not addressed in this thesis.

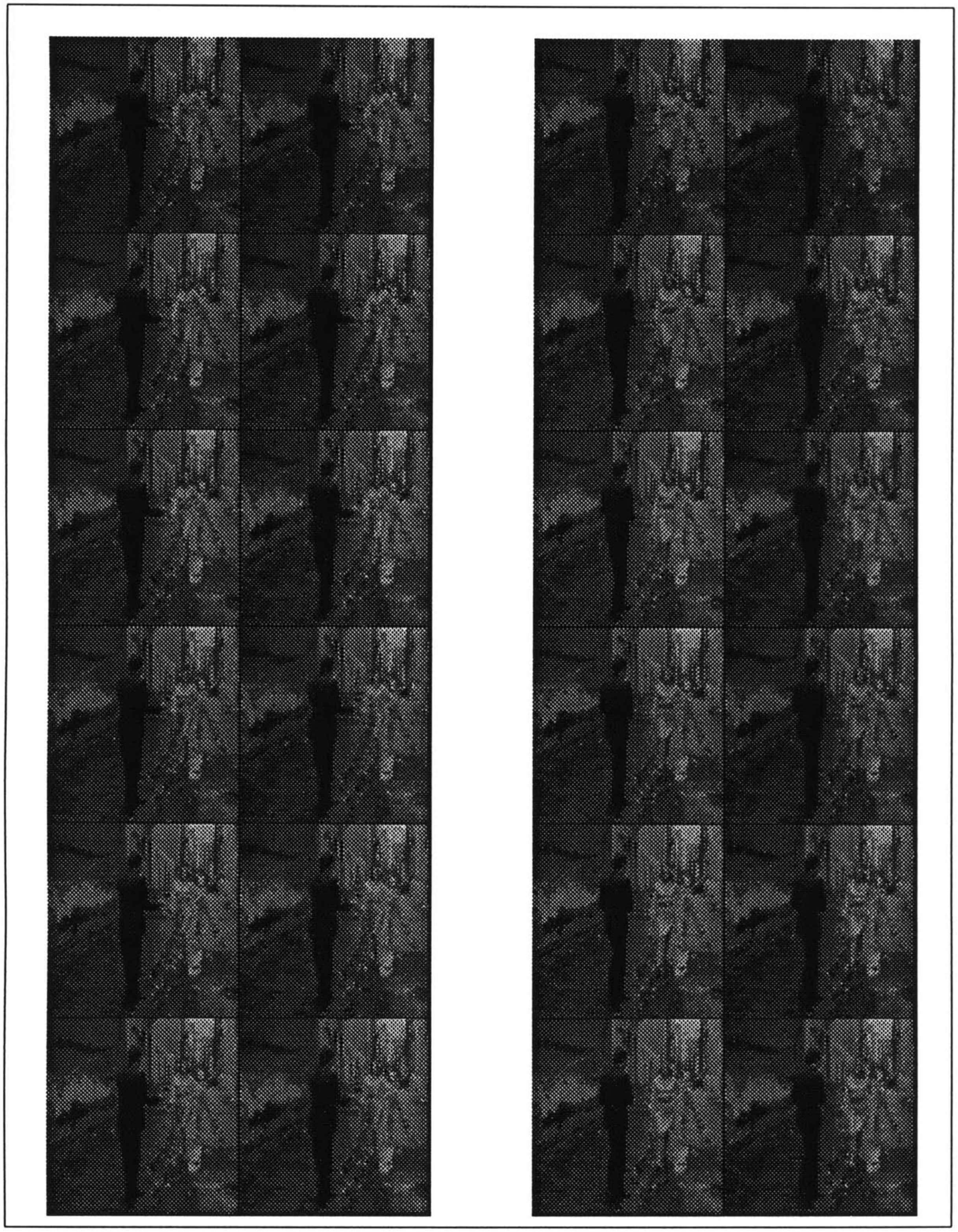

Figure 5-1: Motion sequence coded at 64Kbit / sec. The left image in each column shows the original 128x128 image, the right image is the coded version. Approximately one second of video is shown.

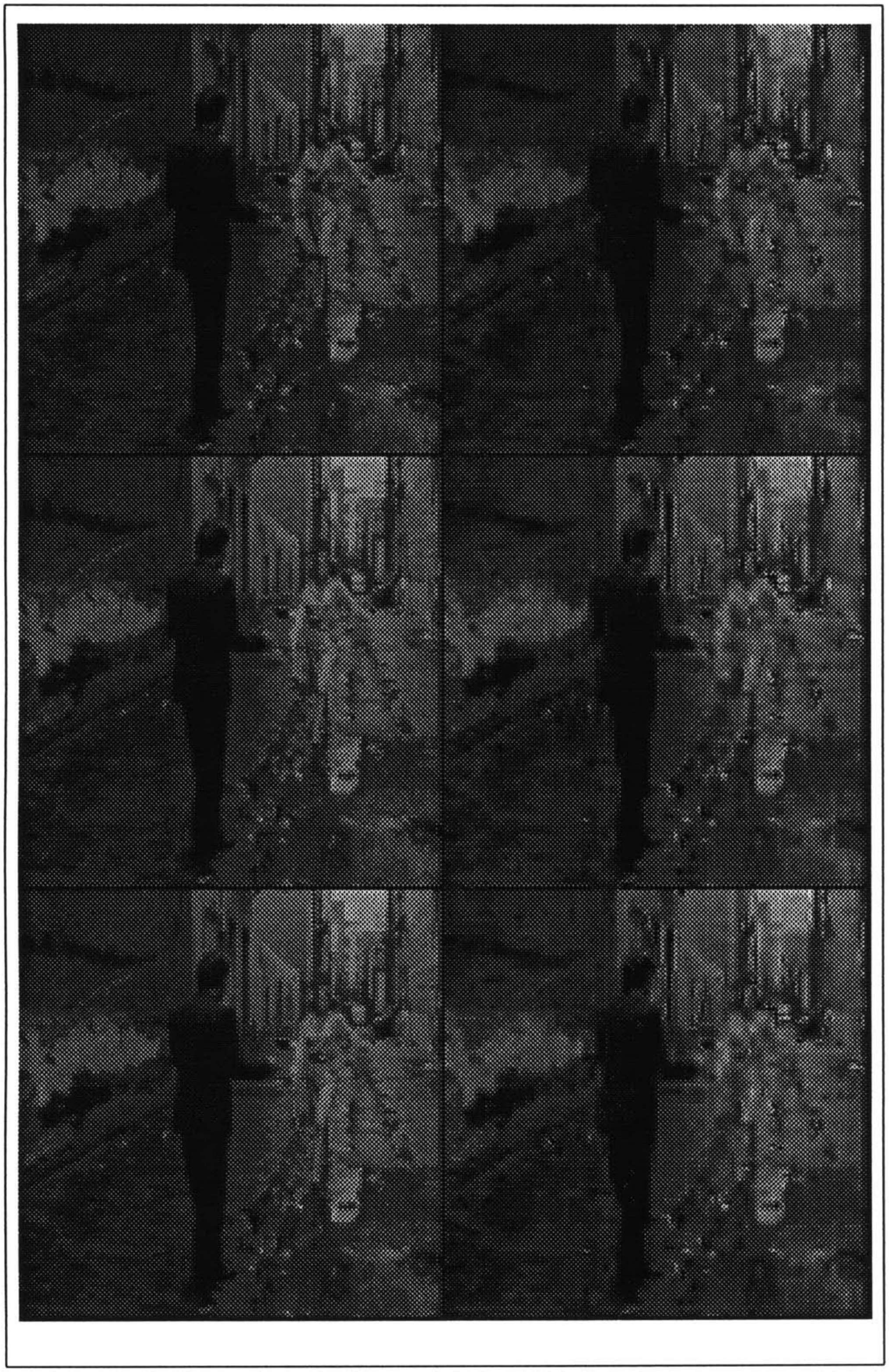

Figure 5-2: First three frames of motion sequence enlarged to show detail.

Semantic Image Representations:

The Recovery of Non-Rigid Motion and Structure

Chapter 6

Interpretation of Non-Rigid Motion

The previous section concluded with a discussion of coding image sequences using a simple frame to frame differencing method. More sophisticated coders use a technique known as motion compensation to predict how objects in the scene are spatially displaced from frame to frame. These techniques are inherently 2-D, and most assume object motion is confined to translation in the image plane [3]. The remainder of this thesis addresses the problem of *3-D* motion estimation from 2-D data. This information could be integrated with the techniques described in the previous section to form a sophisticated coding system, although that endeavor is beyond the scope of this thesis. More importantly, we recover a high-level semantic description of the scene which could be used for content-based coding as well as image-level coding.

6.1 Introduction

To date almost all research on recovering structure from optical flow has been based on rigid motion, either of surface patches or of 3-D structures. Even schemes which address non-rigid motion, e.g., Ullman's incremental rigidity scheme [32], are usually based on minimizing the deviation from a rigid-body interpretation.

Yet non-rigid motion is everywhere: trees sway, flags flap, fish wriggle, arm and leg muscles bunch up, neck and body twist and bend, and cheeks bulge and stretch. One way of coping with nonrigidity is to simply abandon the idea of recovering a whole-body description of the motion, and seek to recover structure on a patch-by-patch basis, perhaps allowing some limited form of non-rigidity [34, 29]. Unfortunately, using a faceted approximation requires that we limit ourselves to using only noise-sensitive local measurements, and that we correctly and consistently segment optical flow into the same facets or patches. Consequently such patch-by-patch recovery of structure is not likely to be either very accurate or robust.

Moreover, we *want* more than a patch-by-patch description, because whole-body motions like bending, twisting and the like are meaningful [25, 19], especially when trying to interpret the actions and gestures of animals and people. To quote Gibson [11]: "An elastic motion, including that of a walking man with his gestures and facial expressions, could be analyzed into a set of rigid motions of elementary particles if one wished to do so, but it is better thought of in terms of components like bending, flexing, stretching, skewing, expanding, and bulging." Recovering such descriptions is exactly the goal of this paper.

We suggest that the main limitation of previous approaches to non-rigid motion was that non-rigid motion was conceptualized as being completely unstructured. As a consequence all one can say about it is how each point or patch is moving. To describe such completely unstructured motion requires three unknowns per object point, and as a consequence the problem of estimating non-rigid motions becames badly underconstrained. In fact, however, most real objects are made of approximately elastic materials, and we will show this fact can be used to transform the non-rigid motion problem in to an *over*constrained problem with a reliable and efficient solution.

The key insight is that the coherent, elastic behavior of real materials implies that non-rigid, whole-body motion can be accurately described with relatively few parameters.[1] The optimal parameterization is obtained from the eigenvectors of the object's corresponding finite element method (FEM) model. These eigenvectors are often referred to as the 3-D object's *free vibration* or *deformation* modes. The parameterization is unique, and is obtained by a multi-scale orthonormal linear transform (similar to the Fourier transform) that maps the object's point-by-point motion in Cartesian coordinates into a coordinate system based on the object's intrinsic deformation modes.

By describing object behavior using a truncated series of vibration/deformation modes one can obtain the best RMS error description possible for a given number of parameters. By varying

[1] Except in unusual cases, for example, an object being torn apart by a chaotic flow

the number of description parameters (often as a function of the number of sensor measurements available) one can smoothly make the transition from a coarse qualitative description to a finely detailed, accurate description — just as one can smoothly obtain more accuracy by adding more terms to a Fourier series. The important consequence for the problem of recovering non-rigid motion is that the problem can always be made overconstrained by reducing the number of vibration/deformation modes. The limiting case is rigid-body motion, which is equivalent to using only the lowest six vibration/deformation modes.

Because the modal representation is derived from the ideas of finite element analysis, we will begin by reviewing the finite element method.

6.2 Representation

The finite element method (FEM) is the standard engineering technique for simulating the dynamic behavior of an object. Use of the similar technique of finite differences has become quite popular in machine vision, following the seminal work of Terzopoulos, Witkin, and Kass [30]. One motivation for using the FEM for vision is that vision is often concerned with estimating changes in position, orientation, and shape, quantities that the FEM accurately describes. Another motivation is that by allowing the user to specify forces that are a function of sensor measurements, the intrinsic dynamic behavior of the FEM can be used to solve fitting, interpolation, or correspondence problems.

In the FEM, interpolation functions are developed that allow continuous material properties, such as mass and stiffness, to be integrated across the region of interest. Note that this is quite different from the finite difference schemes commonly used in computer vision, as is explained in Appendix A of Pentland and Sclaroff [23], although the resulting equations are quite similar. One major difference between the FEM and the finite difference schemes is that the FEM provides an analytic characterization of the surface between nodes or pixels, whereas finite difference methods do not. All of the results presented in this paper will be applicable to both the finite difference and finite element formulations.

Having formulated the appropriate FEM integrals, they are then combined into a description in terms of discrete *nodal points*. Energy functionals are then formulated in terms of nodal displacements $\mathbf{U}$, and the resulting set of simultaneous equations is iterated to solve for the nodal displacements as a function of impinging loads **R**:

$$\mathbf{M}\ddot{\mathbf{U}} + \mathbf{C}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} = \mathbf{R} \tag{6.1}$$

where **U** is a $3n$ x 1 vector of the $(\Delta x, \Delta y, \Delta z)$ displacements of the n nodal points relative to the object's center of mass, **M**, **C** and **K** are $3n$ by $3n$ matrices describing the mass, damping, and material stiffness between each point within the body, and **R** is a $3n$ x 1 vector describing the x, y, and z components of the forces acting on the nodes.

Equation 6.1 is known as the *governing equation* in the finite element method, and may be interpreted as assigning a certain mass to each nodal point and a certain material stiffness between nodal points, with damping being accounted for by dashpots attached between the nodal points. Inertial and centrifugal effects are accounted for by adding appropriate off-diagonal terms to the mass matrix. For additional detail about the finite element formulation see [23], or references [13] or [27].

The most obvious drawback of both the finite difference or finite element methods is the large computational expense. Such methods require roughly $3nm_k$ operations per time step, where $3n$ is the order of the stiffness matrix and m_k is its half bandwidth.[2] Normally $3n$ such time steps are required to obtain an equilibrium solution. For a full 3-D model, where typically $m_k \approx 3n/2$, the computational cost scales as $O(n^3)$. Because of this poor scaling behavior, equations are sometimes discarded (for example, those equations corresponding to internal nodes, as in [30]) in order to obtain sparse banded matrices. In this case the computational expense is reduced to "only" $O(n^2 m_k)$.

A related drawback in vision applications is that the number of description parameters is often roughly equal to the number of sensor measurements, necessitating the use of heuristics such as symmetry and smoothness. This results in non-unique and unstable descriptions, with the consequence that it is difficult to determine whether or not two models are equivalent.

Perhaps the most important problem with using such physically-based methods for vision, however, is that all of the degrees of freedom are coupled together. Thus closed form solutions are impossible, and solutions to the sort of inverse problems encountered in vision are very

[2] See Bathe[13] Appendix A.2.2 for complete discussion on bandwidth of a stiffness matrix.

difficult.

Thus there is a need for a method which transforms the Equation 6.1 into a form which is not only less costly, but also admits of closed-form solutions. Since the number of operations is proportional to the half bandwidth m_k of the stiffness matrix, a reduction in m_k will greatly reduce the cost of step-by-step solution. Moreover, if we can actually diagonalize the system of equations, then the degrees of freedom will become uncoupled and we will be able to find closed-form solutions.

6.2.1 Modal Analysis

To accomplish the goal of diagonalizing the system of equations a linear transformation of the nodal point displacements **U** can be used:

$$\mathbf{U} = \mathbf{P}\tilde{\mathbf{U}} \tag{6.2}$$

where **P** is a square orthogonal transformation matrix and $\tilde{\mathbf{U}}$ is a vector of generalized displacements. Substituting Equation 6.2 into Equation 6.1 and premultipling by $\mathbf{P}^T$ yields:

$$\tilde{\mathbf{M}}\ddot{\tilde{\mathbf{U}}} + \tilde{\mathbf{C}}\dot{\tilde{\mathbf{U}}} + \tilde{\mathbf{K}}\tilde{\mathbf{U}} = \tilde{\mathbf{R}} \tag{6.3}$$

where

$$\tilde{\mathbf{M}} = \mathbf{P}^T\mathbf{M}\mathbf{P}; \quad \tilde{\mathbf{C}} = \mathbf{P}^T\mathbf{C}\mathbf{P}; \quad \tilde{\mathbf{K}} = \mathbf{P}^T\mathbf{K}\mathbf{P}; \quad \tilde{\mathbf{R}} = \mathbf{P}^T\mathbf{R} \tag{6.4}$$

With this transformation of basis set a new system of stiffness, mass and damping matrices can be obtained which has a smaller bandwidth then the original system.

Use of Free Vibration Modes

The optimal transformation matrix **P** is derived from the free vibration modes of the equilibrium equation. Beginning with the governing equation, an eigenvalue problem can be derived

$$\mathbf{K}\phi_i = \omega_i^2\phi_i\mathbf{M} \tag{6.5}$$

which will determine an optimal transformation basis set.

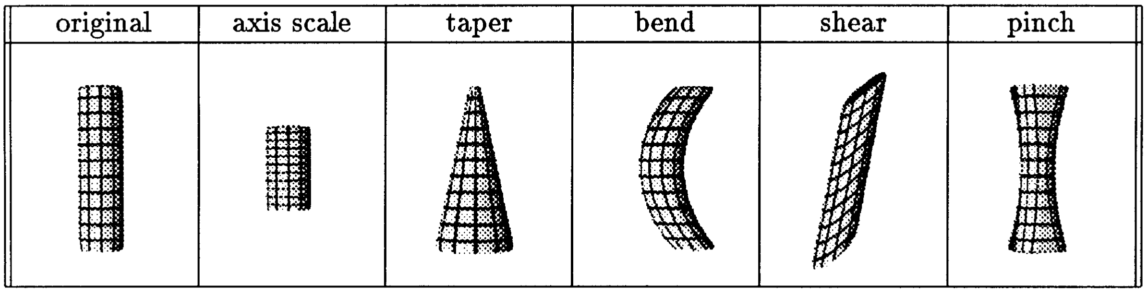

Figure 6-1: Several of the lowest-frequency vibrations modes of a cylinder.

The eigenvalue problem in Equation 6.5 yields $3n$ eigensolutions

$$(\omega_1^2, \phi_1), (\omega_2^2, \phi_2), \ldots, (\omega_n^2, \phi_{3n})$$

where all the eigenvectors are $\mathbf{M}$-orthonormalized. Hence

$$\phi_i^T \mathbf{M} \phi_j \begin{cases} = & 1; \quad i = j \\ = & 0; \quad i \neq j \end{cases} \tag{6.6}$$

and

$$0 \leq \omega_1^2 \leq \omega_2^2 \leq \omega_3^2 \leq \ldots \leq \omega_{3n}^2 \tag{6.7}$$

The eigenvector $\phi_{\mathbf{i}}$ is called the i^{th} mode's *shape vector* and ω_i is the corresponding frequency of vibration. Each eigenvector $\phi_{\mathbf{i}}$ consists of the (x, y, z) displacements for each node, that is, the $3j-2$, $3j-1$, and $3j$ elements are the x, y, and z displacements for node j, $1 \leq j \leq n$.

The lowest frequency modes are always the rigid-body modes of translation and rotation. The eigenvector corresponding to x-axis translation, for instance, has ones for each node's x-axis displacement element, with all other elements being zero. In the finite element formulation rotational motion is linearized, so that nodes on the opposite sides of the body have opposite directions of displacement.

The next-lowest frequency modes are smooth, whole-body deformations that leave the center of mass and rotation fixed. That is, the (x, y, z) displacements of the nodes are a low-order function of the node's position, and the nodal displacements balance out to zero net translational and rotational motion. Compact bodies (simple solids whose dimensions are within the same order of magnitude) normally have low-order modes that are similar to those shown in Figure 6-1. Bodies with very dissimilar dimensions, or which have holes, etc., can have substantially more complex low-frequency modes.

Using these modes we can define a transformation matrix $\mathbf{\Phi}$, which has for its columns the eigenvectors ϕ_i, and a diagonal matrix $\mathbf{\Omega}^2$, with the eigenvalues ω_i^2 on its diagonal:

$$\mathbf{\Phi} = [\phi_1,\ \phi_2,\ \phi_3,\ \ldots,\ \phi_{3n}] \tag{6.8}$$

$$\mathbf{\Omega}^2 = \begin{bmatrix} \omega_1^2 & & & & & \\ & \omega_2^2 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & \omega_{3n}^2 \end{bmatrix} \tag{6.9}$$

Using (6.9) Equation (6.5) can now be written as:

$$\mathbf{K\Phi} = \mathbf{\Omega}^2\mathbf{\Phi M} \tag{6.10}$$

and since the eigenvectors are $\mathbf{M}$-orthonormal:

$$\mathbf{\Phi}^T\mathbf{K\Phi} = \mathbf{\Omega}^2, \qquad \mathbf{\Phi}^T\mathbf{M\Phi} = \mathbf{I} \tag{6.11}$$

From the above formulations it becomes apparent that matrix $\mathbf{\Phi}$ is the optimal transformation matrix $\mathbf{P}$ for systems in which damping effects are negligible.

When the damping matrix $\mathbf{C}$ is restricted to be *Rayleigh damping*, then it is also diagonalized by this transformation. Restriction to this form is equivalent to the assumption that damping, which describes the overall energy dissipation during the system response, is proportional to system response. For this reason Rayleigh damping is commonly assumed in finite element analysis [13].

In summary, we have shown that the general finite element governing equation is decoupled when using a transformation matrix $\mathbf{P}$ whose columns are the free vibration mode shapes of the FEM system [13, 27, 24, 20]. These decoupled equations may then be integrated numerically (see [24]) or solved in closed form by use of a *Duhamel* integral (see [20]).

6.2.2 Accuracy and The Number of Modes Employed

The modal representation decouples the degrees of freedom within the non-rigid dynamic system of Equation 6.1, but it does not by itself reduce the total number of degrees of freedom. However modes associated with high resonance frequencies (large eigenvalues) normally have little effect on object shape. This is because for a given excitation energy the displacement amplitude for each mode is *inversely* proportional to the *square* of the mode's resonance frequency, and because damping is proportional to a mode's frequency. The consequence of these combined effects is that high-frequency modes generally have very little amplitude.

We can therefore discard high-frequency modes with little loss of accuracy or generality, with the result that we have fewer equations in fewer unknowns and (because of Nyquist considerations) we can employ a much larger time step when performing a simulation. For a typical problem this approach can decrease computational cost by *two orders of magnitude* while maintaining good accuracy [24]. Moreover, for a fixed number of modes the computation scales as $O(n)$ rather than $O(n^2)$ or $O(n^3)$. For complex shapes this linear scaling behavior can be extremely important, and it is for this reason that this type of reduced-basis modal analysis has become the standard method for extremely large engineering problems, such as the analysis of airplane frames or large buildings.

Figure 6-2 shows a sampling of shapes that can be achieved by elastically deforming an initial spherical shape using its 30 lowest-frequency deformation modes (note that the first six modes are rigid-body translation and rotation). As can be seen, a wide range of non-rigid motions and their resulting shapes can be produced with relatively few deformation modes. As Figure 6-2 illustrates, discarding high-frequency modes is **not** equivalent to assuming that the surface is smooth, because we can still generate sharp bends, creases, and so forth.

As can be seen, besides decoupling the degrees of freedom the modal representation also provides a natural hierarchy of scale, so that we can smoothly vary the level of detail by adding in or discarding high-frequency modes. That is, the modal representation provides a natural

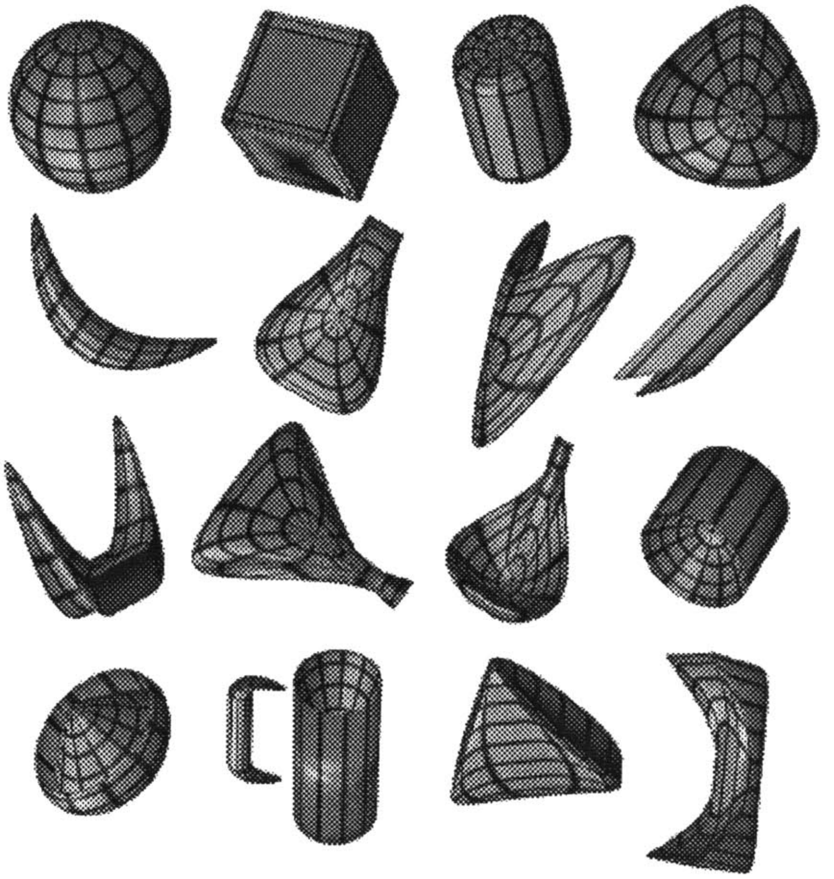

Figure 6-2: A sampling of shapes produced by elastically deforming an initial spherical shape using its 30 lowest-frequency deformation modes. As can be seen, a wide range non-rigid motions and their resulting shapes can be described by relatively few deformation modes.

multi-scale representation for 3-D object shape in much the same manner that the Fourier transform provides a multi-scale representation for images.

The ability to decouple and order the degrees of freedom has several important implications for machine vision applications. For instance, by matching the level of detail (the number of modes) to the number of sensor measurements the shape recovery process can be kept overconstrained. Similarly, because the degrees of freedom are orthogonal, shape and velocity descriptions are *unique* for an object in canonical position (the restriction to canonical position is necessary because rotations have been linearized). These properties contrast markedly to representations based on nodes, polynomials, or splines where descriptions are normally neither unique nor overconstrained.[3]

Because of this uniqueness property, the modal representation is well-suited for object recognition and other spatial database tasks. To compare two objects described using a modal representation one simply compares the vector of mode values $\tilde{\mathbf{U}}$; if the dot product of the $\tilde{\mathbf{U}}$ for each object is small, then the objects are similar (again assuming that the objects are in canonical position).

[3] As the number of nodes used to define the finite element model increases, the underlying continuous differential equations can be more accurately approximated. This is simply because there are more high-frequency modes in the model; assuming that all models correctly capture the geometry the low-frequency modes are not affected by number of nodes defining the finite element model.

Chapter 7

Recovering Motion

7.1 Recovering Motion

In this thesis we will analyze the case where the object geometry at time $t = 0$ is known, and where object motion is viewed under orthographic projection. The problem of obtaining the initial object geometry and vibration mode description is addressed by Pentland and Sclaroff [23]. The problem, then, is to find the rigid and non-rigid 3-D motions $d\mathbf{U}/dt$ that best account for the observed 2-D image velocities. The major difficulty in finding such a solution is that there are $3n$ unknown degrees of freedom in the model and at most $2n$ degrees of freedom in the observations. Thus we must somehow reduce the number of unknowns to obtain a solution.

The modal representation offers a principled, physically-based method for reducing the number of degrees of freedom. Because we know that the elastic properties of real materials imply that the high-frequency modes are (almost always) of low amplitude, we can discard many of these modes without incurring significant error.

Further, because the modal representation is frequency-ordered, it has stability properties that are similar to those of a Fourier decomposition. Just as with the Fourier decomposition, an exact[1] subsampling of the data points points does not change the low-frequency modes. Similarly, irregularities in local sampling and measurement noise tend to primarily affect the high-frequency modes, leaving the low-frequency modes relatively unchanged.

[1] E.g., if there are 2^k points a sampling of every 2^l points $(l < k)$ is an exact subsampling.

Again, we note that discarding these high-frequency modes is **not** equivalent to a general smoothness constraint, because we can still generate sharp bends, creases, and so forth (see Figure 6-2). What we cannot do with a reduced-basis modal representation is generate several sharp creases that are close together.

We will therefore pose our problem in the modal coordinate system: the problem is to find the set of 3-D *mode* velocities $d\tilde{\mathbf{U}}/dt$ that best account for the observed 2-D image velocities. If we use the m lowest frequency modes, then there will be only m unknown degrees of freedom in the model and up to $2n$ degrees of freedom in the observations. Thus by appropriate choice of m the problem can always be made overconstrained. Whenever the 3-D velocities of the individual nodes are required, we can convert $\tilde{\mathbf{U}}$ back to the original space coordinates by multiplying by $\mathbf{\Phi}$.

7.1.1 Kinematic Solution

We first note that ϕ_i, the i^{th} column of $\mathbf{\Phi}$, describes the *deformation* the object experiences as a consequence of the modal force $\tilde{r}_i$. Or, perhaps more intuitively, ϕ_i describes how each of the n nodal points $(x_j, y_j, z_j)^T$ change as a function of $\tilde{u}_i$, the i^{th} mode's amplitude,

$$\phi_i = (\frac{dx_1}{d\tilde{u}_i}, \frac{dy_1}{d\tilde{u}_i}, \frac{dz_1}{d\tilde{u}_i}, \ldots \frac{dx_n}{d\tilde{u}_i}, \frac{dy_n}{d\tilde{u}_i}, \frac{dz_n}{d\tilde{u}_i})^T \quad . \tag{7.1}$$

Letting $\mathbf{V}$ be the 3-D velocity of each node,

$$\mathbf{V} = (\frac{dx_1}{dt}, \frac{dy_1}{dt}, \frac{dz_1}{dt}, \ldots \frac{dx_n}{dt}, \frac{dy_n}{dt}, \frac{dz_n}{dt})^T \quad , \tag{7.2}$$

we then have that

$$\mathbf{V} = \mathbf{\Phi}\frac{d\tilde{\mathbf{U}}}{dt} = \mathbf{\Phi}\dot{\tilde{\mathbf{U}}} \quad . \tag{7.3}$$

Given the 3-D motions of each node, then we can solve for the modal velocities $\dot{\tilde{\mathbf{U}}}$:

$$\dot{\tilde{\mathbf{U}}} = \mathbf{\Phi}^{-1}\mathbf{V} \tag{7.4}$$

Thus having observed 3-D nodal velocities $\mathbf{V}$, the kinematic solution for the modal amplitudes

$\tilde{\mathbf{U}}^t$ at time t is simply

$$\tilde{\mathbf{U}}^t = \dot{\tilde{\mathbf{U}}}^t \Delta t + \tilde{\mathbf{U}}^{t-1} = \mathbf{\Phi}^{-1}\mathbf{V}^t \Delta t \tag{7.5}$$

Note that $\tilde{\mathbf{U}}^{t-1} = \mathbf{0}$, that is, the object's rest state is taken to be its shape at time $t-1$, so that the nodal displacements at time t are calculated relative the the nodal positions at time $t-1$. The primary limitation of this solution stems from the finite element method's linearization of modes such as rotation. Because these modes are linearized, it is important to limit inter-frame motion to small rotations (less than 10°) and deformations (less than 10% of the object size).

7.1.2 Estimation from 2-D Data

Given the kinematic solution of Equation 7.5, the remaining problem is to obtain a generalization that uses two-dimensional measurements of optical flow as input data. More concretely, the problem is to estimate the rigid-body motion and non-rigid 3-D object deformation at each each subsequent time t given only noisy estimates of 2-D (orthographically projected) optical flow (u_i^t, v_i^t) at m image points (x_i, y_i). The image points (x_i, y_i) are **not** assumed to be either dense or uniformly sampled.

This can be accomplished by allocating each of the available optical flow vectors (u_i, v_i) among the nodal points whose image projections are close to (x_i, y_i), the flow vector's image position. The most accurate method of accomplishing this allocation is to use the interpolation functions $\mathbf{H}$ used to define the finite element model (see [23] Appendix A). However when the mesh of nodes is sufficiently dense that each optical flow vector projects near to some node, then we have found that it is sufficiently accurate to use simple bilinear interpolation of the flow vectors to the surrounding three nodes. An inexpensive method of accomplishing this is discussed reference [23] Appendix B.

This produces estimates of the projected 2-D nodal velocities

$$\mathbf{V}_P = (u_1, v_1, u_2, v_2, \ldots u_n, v_n)^T \tag{7.6}$$

We define the matrix $\mathbf{\Phi}_P$ similarly, by removing rows of $\mathbf{\Phi}$ that correspond to z-axis displacements. Note that nodes without nearby optical flow may have no velocity estimate; therefore rows of $\mathbf{V}_P$ and $\mathbf{\Phi}_P$ corresponding to the x and y displacements of these nodes are undefined (contain no information) and must also be removed.

Some modes, including translation, scaling and linear shearing along the z axis, cannot be observed under orthographic projection. Therefore columns of $\mathbf{\Phi}_P$ and rows of $\tilde{\mathbf{U}}$ corresponding to these modes must also be removed. Because the remaining mode shapes are orthogonal to the translation, scaling and shearing modes they remain unaffected. Similarly, in some cases the 2-D motions caused by a particular modal deformation are very small, so that the mode's amplitude cannot be reliably estimated. Such ill-conditioned estimates can be prevented by discarding modes for which the corresponding column of $\mathbf{\Phi}_P$ has small magnitude.

With these definitions we may now generalize Equation 7.5 to obtain an estimate of the object's 3-D shape $\tilde{\mathbf{U}}^t$ at time t based on the optical flow data. The generalization is simply:

$$\tilde{\mathbf{U}}^t = \mathbf{\Phi}_P^{-1}\mathbf{V}_P^t\Delta t \tag{7.7}$$

Equation 7.7 is underconstrained if all of the modes are present in $\mathbf{\Phi}_P$; however, by discarding a sufficient number of the low-amplitude, high-frequency modes, the estimate can always be made overconstrained. [2] Therefore, in practice, $\mathbf{\Phi}_P^{-1}$ is calculated by use of a Moore-Penrose pseudoinverse:

$$\tilde{\mathbf{U}} = \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{V}_P\Delta t \tag{7.8}$$

with the columns of $\mathbf{\Phi}_P$ and the rows of $\tilde{\mathbf{U}}$ corresponding to high-frequency modes deleted.

Equation 7.8 provides us with a least-squares estimate of $\tilde{\mathbf{U}}$, the object's rigid motion and non-rigid deformation. It is the best RMS error estimate of the projected rigid and non-rigid motions given the observed optical flow vectors, where the projected mode shapes are described (analytically) by the columns of $\mathbf{\Phi}_P$ and the finite element interpolation functions $\mathbf{H}$.

We have found that 30 deformation modes are adequate to account for most rigid and non-rigid motions, so that only 15 or so independent flow vectors are required per body. In situations with very sparse flow vectors, we can reduce the number of deformation modes still further in order to keep the calculation overconstrained. In the limiting case, we require only three independent flow vectors in order to estimate the six rigid body motions, and four vectors

[2] Note that when there are many more optical flow vectors than degrees of freedom in the finite element model, the interpolation functions $\mathbf{H}$ act as filters to bandlimit the sensor data, thus reducing aliasing.

to obtain an overconstrained estimate.[3]

7.1.3 Efficiency Considerations for Motion Estimation

When an object rotates the intrinsic (object-centered) and global coordinate systems are no longer identical. Because $\mathbf{\Phi}$ is calculated in the intrinsic coordinate system, it must be either recalculated or modified whenever significant object rotation has occured. The most efficient method is to transform $\mathbf{\Phi}$ to the new coordinate system by rotating the (x, y, z) triplets in the columns of $\mathbf{\Phi}$ by the object's current estimated rotation matrix R:

$$(\phi^*_{i,3*j}, \phi^*_{i,3*j+1}, \phi^*_{i,3*j+2},)^T = R^{-1}(\phi_{i,3*j}, \phi_{i,3*j+1}, \phi_{i,3*j+2},)^T \tag{7.9}$$

for $i = 1, 2, 3, \ldots 3n$ and $j = 1, 2, 3, \ldots n$. Thus as the object rotates the matrix $\mathbf{\Phi}$ is progressively transformed to the object's new intrinsic coordinate system. After each rotation $\mathbf{\Phi}$ is then projected to obtain $\mathbf{\Phi}_P$ as described above, and Equation 7.8 is applied.

[3] Note that this is different than the normal "n views of m points" result in that we are assuming that the initial object geometry is known. Note also the restriction to small inter-frame rotations and deformations.

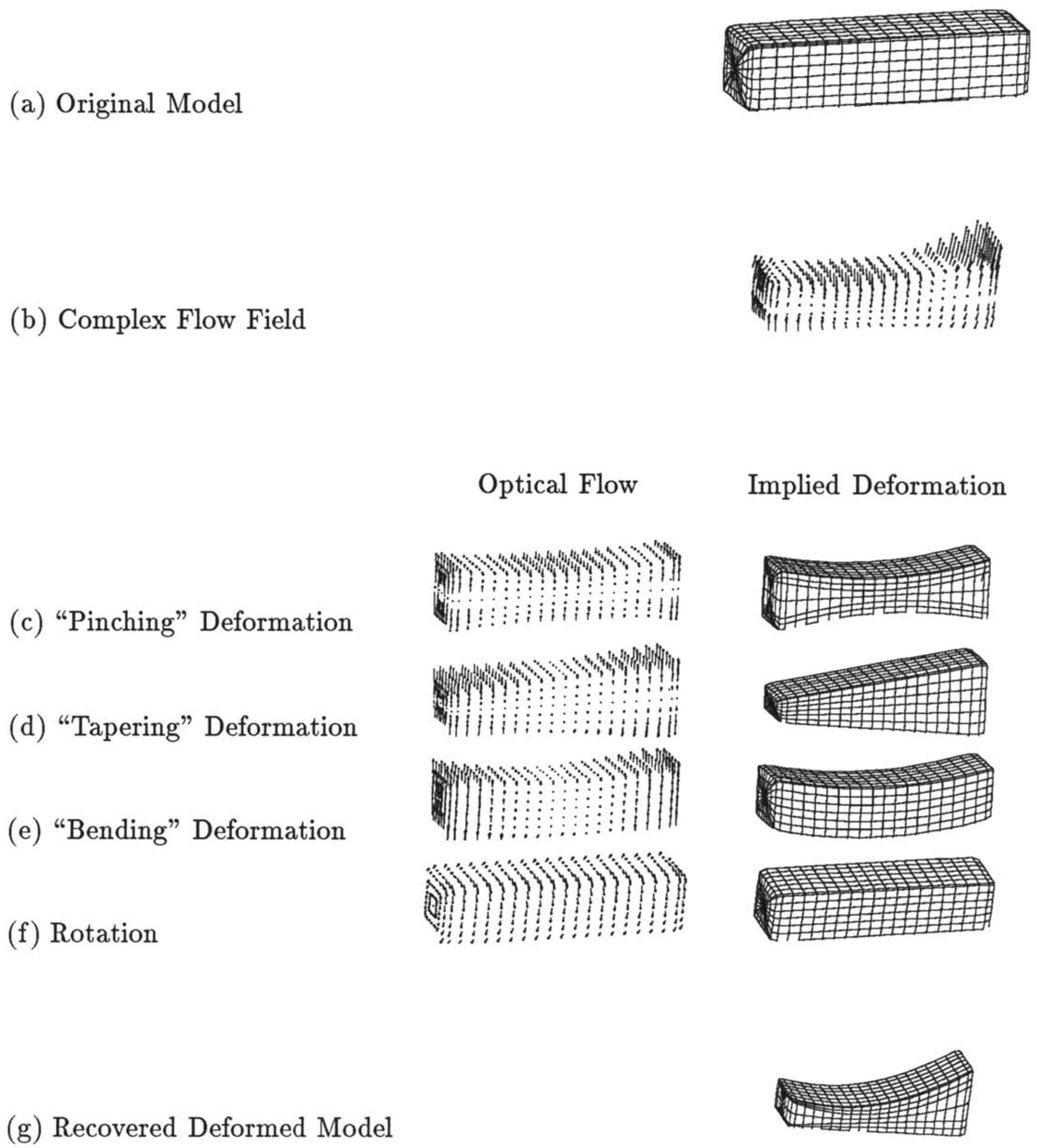
(a) Original Model
(b) Complex Flow Field
Optical Flow
Implied Deformation
(c) "Pinching" Deformation
(d) "Tapering" Deformation
(e) "Bending" Deformation
(f) Rotation
(g) Recovered Deformed Model

Figure 7-1: All motions, including complex non-rigid motions, can be decomposed into the linear sum of a set of orthogonal basis motions. In our algorithm we use the free vibration modes of the object as the basis set. In this example, a complex flow field is decomposed into the sum of three non-rigid motions and a rigid-body rotation.

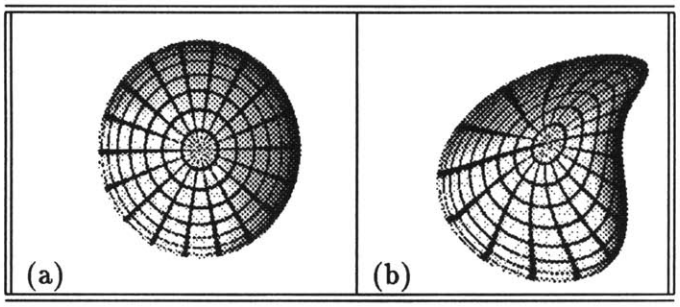

Figure 7-2: Random forces were applied to an elastic spherical body in order to evaluate the accuracy of the kinematic solution. (a) Shape at frame zero, (b) shape at frame one, after a randomly-selected force was applied.

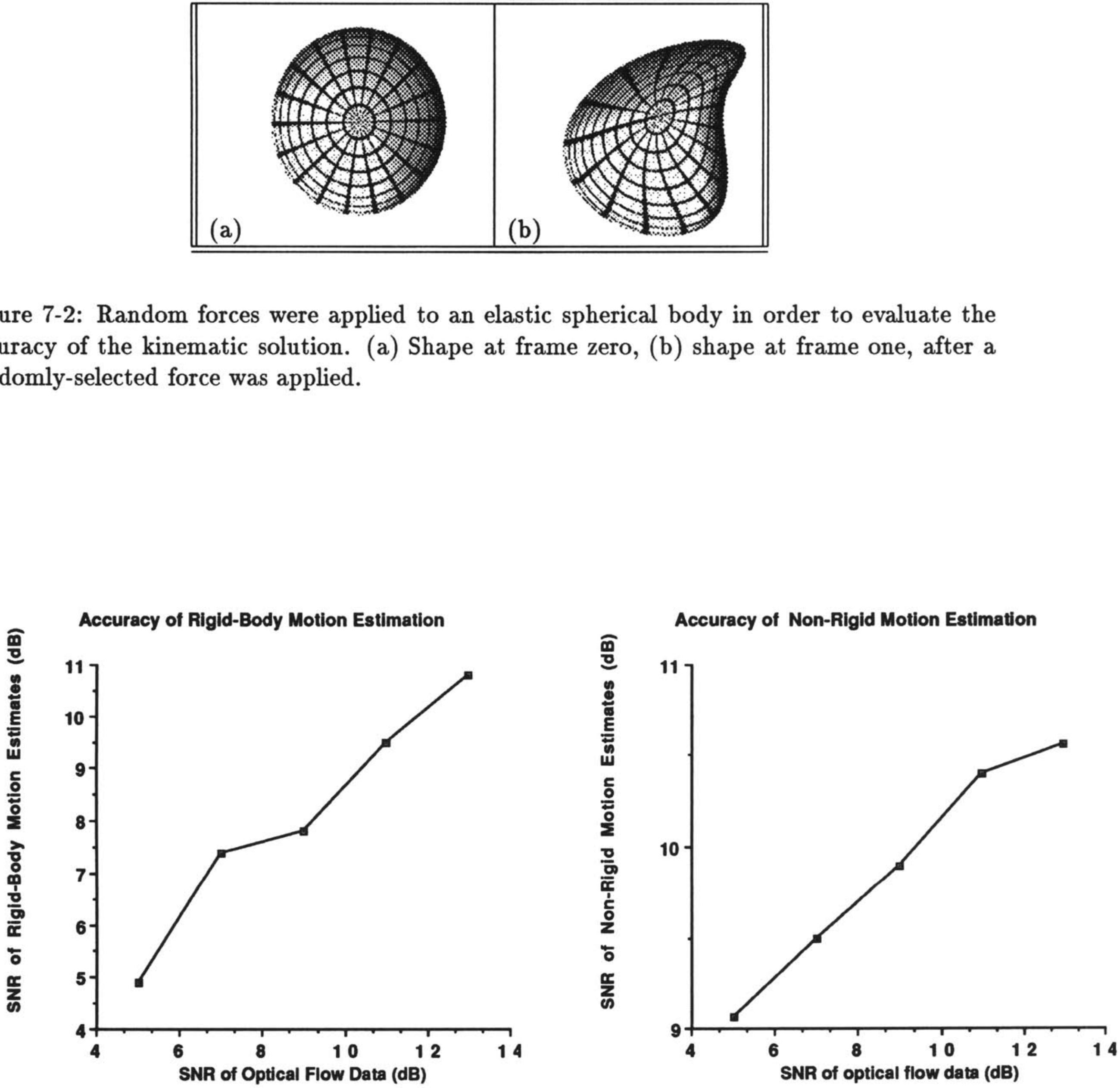

Figure 7-3: (a) Rigid-body motion estimation error, and (b) Non-rigid motion estimation error, both as a function of the SNR of the optical flow field.

Chapter 8

Examples using Synthetic Data

8.1 Examples Using Synthetic Data

8.1.1 An Illustrative Example

Figure 7-1 illustrates how a complex sum of rigid and non-rigid motions can be decomposed into a sum of the object's free vibration modes. Figure 7-1(a) shows a box, and 7-1(b) shows the complex optical flow field generated as the box undergoes both rigid and non-rigid motion. The general approach taken in this paper is to decompose such complex motions into a set of simpler, orthogonal modal deformations. This is accomplished by using Equation 7.8 to "explain" the complex flow field of (b) by use of a set of simpler "modal flow fields" illustrated in Figures 7-1(c-f). The modal amplitudes that are required to obtain this "explanation" provides us with an estimate of the object's new 3-D shape, shown in Figure 7-1(g).

More specifically, the estimation process starts by interpolating the 2-D flow vectors to the nearest nodes on the (assumed known) 3-D model of the box. Note that these flow vectors do **not** have to be dense or regularly sampled. Equation 7.8 is then used to decompose this flow field into a sum of the rigid and non-rigid deformations ("modal flow fields") described by the columns of $\mathbf{\Phi}_P$. These columns correspond to the free vibration modes of the box, as determined from its finite element description. In this case Equation 7.8 determines that three non-rigid and one rigid-body motion have occurred: the non-rigid motions are the pinching-, tapering-, and bending-like motions shown in Figure 7-1(c), (d), and (e), and the rigid-body motion is the rotation shown in Figure 7-1(f).

The combination of these motions, when applied to the original box shape, produces an estimate of object's new shape which is shown in Figure 7-1(g). The result of this decomposition process, therefore, allows us to track, predict, and describe the 3-D rigid and non-rigid motion using only sparse 2-D estimates of optical flow.

8.1.2 A Statistical Evaluation

To evaluate the stability and accuracy of the decomposition and estimation process, an experiment was conducted in which randomly selected forces were applied to an elastic spherical body to produce both rigid-body and non-rigid motions. A typical example is shown in Figure 7-2. The resulting 2-D optical flow field was then observed, and the rigid and non-rigid motions estimated by use of Equation 7.8. To make the experiment more realistic, various amounts of uniformly distributed noise was added to the optical flow field before Equation 7.8 was applied. Each noise condition was repeated with 100 different randomly selected forces and consequent motions. The mean accuracy of the estimation process was then measured.

Figure 7-3 shows the accuracy of the decomposition and estimation process. The signal-to-noise ratio (SNR) for the motion estimates were calculated by

$$\mathrm{dB} = 10 * \log_{10}\left(\frac{\sum \|\tilde{\mathbf{U}}\|^2}{\sum \|\hat{\tilde{\mathbf{U}}} - \tilde{\mathbf{U}}\|^2}\right) \tag{8.1}$$

where $\hat{\tilde{\mathbf{U}}}$ is the estimated motion, $\tilde{\mathbf{U}}$ the true motion, and the sums are taken over the 100 experimental trials. This statistic compares the variance (squared magnitude) of the estimation errors the variance of the true motion.

The SNR of the optical flow field was calculated by

$$\mathrm{dB} = 10 * \log_{10}\left(\frac{\sum \|\mathbf{V}_P\|^2}{\sum \|\hat{\mathbf{V}}_P - \mathbf{V}_P\|^2}\right) \tag{8.2}$$

where $\hat{\mathbf{V}}_P$ is the noise-corrupted 2-D flow field, $\mathbf{V}_P$ the true 2-D flow field, and the sums are taken over the 100 experimental trials. This statistic compares the variance of the flow field noise to the variance of the true flow field.

Figure 7-3(a) shows the accuracy at estimating rigid-body motion as a function of the signal-to-noise ration (SNR) of the optical flow field (i.e., only the first six elements of $\tilde{\mathbf{U}}$, which are the rigid-body modes, were used to compute the SNR). Although only rigid-body accuracy is shown here, both rigid-body and non-rigid motions were estimated simultaneously. It can be seen that the accuracy of estimation is linearly related to the SNR of the flow field. The most noisy condition shown here (approximately 5 dB SNR) corresponds to approximately 56% added noise.

Figure 7-3(b) shows the accuracy at estimating non-rigid motions as a function of the signal-to-noise ratio (SNR) of the optical flow field (i.e., elements 7 through 30 of $\tilde{\mathbf{U}}$, which are the non-rigid motion modes, were used to compute the SNR). Although only non-rigid accuracy is shown here, both rigid-body and non-rigid motions were estimated simultaneously. Again, it can be seen that the accuracy of estimation is linearly related to the SNR of the flow field up to at least 50% noise.

The major factor that permits the stability and noise-resistance shown here is the fact that data is integrated over the entire body rather than only over a small patch. It should be noted, however, that because $\mathbf{\Phi}$ linearizes object rotation and deformation it is important that inter-image rotations and deformations remain small. For larger rotations and deformations we have found that it is necessary to use an iterative estimation scheme.

Chapter 9

Kalman Filtering for Dynamic Motion Estimation

In the previous sections we have addressed kinematic estimation, where velocity at only one instant is considered. For time sequences, however, it is necessary to also consider the *dynamic* properties of the body and of the data measurements. The Kalman filter [14, 15] is the standard technique for obtaining estimates of the state vectors of dynamic models, and for predicting the state vectors at some later time. Outputs from the Kalman filter are the optimal (weighted) least-squares estimate for non-Gaussian noises [2, 9].

The first use of Kalman filtering for motion estimation was by Brodia and Chellappa [6], who presented a careful evaluation of the approach. Work by Faugeras, Ayache and their collegues, and more recently many others, has thoroughly developed the subject [8, 4]. In this section we will develop a Kalman filter that estimates position and velocity for the finite element modal parameters. We will then show that this particular type of Kalman filter is mathematically equivalent to time integration of the FEM governing equation for appropriate choices of mass **M** and stiffness **K**. That is, the Kalman filter may be viewed as a simulation of the model's behavior, with the observed optical flow acting as guiding "forces."

9.0.3 The Kalman Filter

Let us define a dynamic process

$$\dot{\mathbf{X}} = \mathbf{A}\mathbf{X} + \mathbf{B}a \tag{9.1}$$

and observations

$$\mathbf{Y} = \mathbf{C}\mathbf{X} + n \tag{9.2}$$

where a and n are white noise processes having known spectral density matrices. Then the *optimum observer* [14, 15] is given by the following *Kalman filter*

$$\dot{\hat{\mathbf{X}}} = \mathbf{A}\mathbf{X} + \hat{\mathbf{K}}_f(\mathbf{Y} - \mathbf{C}\mathbf{X}) \tag{9.3}$$

provided that the Kalman gain matrix $\hat{\mathbf{K}}_f$ is chosen correctly.

The Kalman Gain Factor

The gain matrix $\hat{\mathbf{K}}_f$ in Equation 9.3 minimizes the covariance matrix $\mathbf{P}$ of the error $\mathbf{e} = \mathbf{X} - \hat{\mathbf{X}}$. Assuming that the cross-variance between the system excitation noise a and the observation noise n is zero, then

$$\hat{\mathbf{K}}_f = \mathbf{P}\mathbf{C}^T\mathcal{N}^{-1} \tag{9.4}$$

where the observation noise spectral density matrix $\mathcal{N}$ must be nonsingular [9]. Assuming that the noise characteristics are constant, then the optimizing covariance matrix $\mathbf{P}$ is obtained by solving the differential of $\mathbf{P}$

$$0 = \dot{\mathbf{P}} = \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^T - \mathbf{P}\mathbf{C}^T\mathcal{N}^{-1}\mathbf{C}\mathbf{P} + \mathbf{B}\mathcal{A}\mathbf{B}^T \tag{9.5}$$

which is known as the *Riccati equation.*

Estimation of Displacement and Velocity

In the current application we are primarily interested in estimation of the modal amplitudes $\tilde{\mathbf{U}}$ and their velocities $\tilde{\mathbf{V}} = \dot{\tilde{\mathbf{U}}}$. $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ are therefore the state variables of our dynamic system, and the governing equations are

$$\begin{aligned} \dot{\tilde{\mathbf{U}}} &= \tilde{\mathbf{V}} \\ \dot{\tilde{\mathbf{V}}} &= \mathrm{a} \end{aligned} \tag{9.6}$$

where a is a vector of externally-applied nodal accelerations, which will be considered to be noise. The observed variable will be $\mathbf{V}_P$, the 2-D nodal velocities. Then from Equation 7.7 we have that

$$\mathbf{V}_P = \frac{\mathbf{\Phi}_P}{\Delta t}\tilde{\mathbf{U}} + n \tag{9.7}$$

where n is a vector giving the observation noise, and again the last estimate of shape is taken to be the object's rest state, e.g., $\tilde{\mathbf{U}}^{t-1} = \mathbf{0}$.

It is important to note that both rotational and non-rigid dynamics are inherently non-linear. The finite element formulation, however, linearizes this behavior so that small times stepsand small nodal displacements are required to obtain an accurate simulation using the FEM. Because we are employing the FEM's linearization of dynamic behavior for motion estimation, our formulation will be an *extended* Kalman filter. The behavior of such an extended Kalman filter is difficult to analyze mathematically, and so properties such as convergence and unbiased estimation must be evaluated experimentally or numerically. Such an evaluation will be presented in the following section.

In state-space notation the system of equations is

$$\begin{bmatrix} \dot{\tilde{\mathbf{U}}} \\ \dot{\tilde{\mathbf{V}}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{U}} \\ \tilde{\mathbf{V}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \mathrm{a} \tag{9.8}$$

where $\mathbf{I}$ is the n x n identity matrix.

Comparing with Equations 9.1 and 9.2 we obtain

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad , \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \quad , \quad \mathbf{C}^T = \begin{bmatrix} \mathbf{\Phi}_P/\Delta t \\ \mathbf{0} \end{bmatrix}$$

The Kalman filter is therefore

$$\begin{bmatrix} \dot{\tilde{\mathbf{U}}} \\ \dot{\tilde{\mathbf{V}}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{U}} \\ \tilde{\mathbf{V}} \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{K}}_{f,1} \\ \hat{\mathbf{K}}_{f,2} \end{bmatrix} \left(\mathbf{V}_P - [\mathbf{\Phi}_P/\Delta t \quad \mathbf{0}] \begin{bmatrix} \tilde{\mathbf{U}} \\ \tilde{\mathbf{V}} \end{bmatrix} \right) \tag{9.9}$$

where $\hat{\mathbf{K}}_{f,1}$ and $\hat{\mathbf{K}}_{f,2}$ are the Kalman gain matrices for velocity and acceleration, respectively. By collapsing terms, rewriting $\dot{\tilde{\mathbf{U}}} = \tilde{\mathbf{V}}$, Equation 9.9 becomes

$$\begin{bmatrix} \dot{\tilde{\mathbf{U}}} \\ \ddot{\tilde{\mathbf{U}}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{\mathbf{U}}} + \hat{\mathbf{K}}_{f,1} \left(\mathbf{V}_P - \mathbf{\Phi}_P \tilde{\mathbf{U}}/\Delta t \right) \\ \hat{\mathbf{K}}_{f,2} \left(\mathbf{V}_P - \mathbf{\Phi}_P \tilde{\mathbf{U}}/\Delta t \right) \end{bmatrix} \tag{9.10}$$

We may solve for the Kalman gain matrices by first using Equation 9.4 to obtain

$$\begin{bmatrix} \hat{\mathbf{K}}_{f,1} \\ \hat{\mathbf{K}}_{f,2} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{12} & \mathbf{P}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_P/\Delta t \\ 0 \end{bmatrix} \mathcal{N}^{-1} = \begin{bmatrix} \mathbf{P}_{11}\mathbf{\Phi}_P\mathcal{N}^{-1}/\Delta t \\ \mathbf{P}_{12}\mathbf{\Phi}_P\mathcal{N}^{-1}/\Delta t \end{bmatrix} \tag{9.11}$$

where the $\mathbf{P}_{ij}$ are n x n blocks of the error covariance matrix P, and $\mathcal{N}$ is the n x n spectral density matrix of the observation noise n.

We will assume that n and a originate from independent noise with standard deviations n and a respectively. As each point-wise measurement error is spread to the various modes by $\mathbf{\Phi}_P/\Delta t$, it is reasonable to choose $\mathcal{N} = n^2\mathbf{\Phi}_P^2/\Delta t^2$. Using this spectral density matrix the optimum covariance matrix $\mathbf{P}$ can be found by solving Equation 9.5 for the $\mathbf{P}_{ij}$, which yeilds

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 2\mathbf{P}_{12} - n^{-2}\mathbf{P}_{11}^2 & \mathbf{P}_{22} - n^{-2}\mathbf{P}_{11}\mathbf{P}_{12} \\ \mathbf{P}_{22} - n^{-2}\mathbf{P}_{12}\mathbf{P}_{11} & a^2\mathbf{I} - n^{-2}\mathbf{P}_{12}^2 \end{bmatrix} \tag{9.12}$$

Similarly, as a is assumed independent for each mode, then its spectral density matrix is $\mathcal{A} = a^2\mathbf{I}$ and thus

$$\begin{aligned} \mathbf{P}_{11} &= (2an^3)^{1/2}\mathbf{I} \\ \mathbf{P}_{12} &= an\mathbf{I} \\ \mathbf{P}_{22} &= (2a^3n)^{1/2}\mathbf{I} \end{aligned} \tag{9.13}$$

Finally, from Equation 9.4, we can determine the Kalman gain matrices

$$\begin{bmatrix} \hat{\mathbf{K}}_{f,1} \\ \hat{\mathbf{K}}_{f,2} \end{bmatrix} = \begin{bmatrix} (\frac{2a}{n})^{1/2}\mathbf{\Phi}_P^{-1}\Delta t \\ (\frac{a}{n})\mathbf{\Phi}_P^{-1}\Delta t \end{bmatrix} \tag{9.14}$$

Substituting this result into Equation 9.10 we obtain

$$\begin{bmatrix} \dot{\tilde{\mathbf{U}}} \\ \ddot{\tilde{\mathbf{U}}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{\mathbf{U}}} + (\frac{2a}{n})^{1/2}\mathbf{\Phi}_P^{-1}\Delta t \left(\mathbf{V}_P - \mathbf{\Phi}_P/\Delta t\tilde{\mathbf{U}}\right) \\ (\frac{a}{n})\mathbf{\Phi}_P^{-1}\Delta t \left(\mathbf{V}_P - \mathbf{\Phi}_P/\Delta t\tilde{\mathbf{U}}\right) \end{bmatrix} \tag{9.15}$$

Letting $\tilde{\mathbf{V}}_P = \mathbf{\Phi}_P^{-1}\mathbf{V}_P$, we obtain in the modal coordinate system,

$$\begin{bmatrix} \dot{\tilde{\mathbf{U}}} \\ \ddot{\tilde{\mathbf{U}}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{\mathbf{U}}} + (\frac{2a}{n})^{1/2}\left(\tilde{\mathbf{V}}_P\Delta t - \tilde{\mathbf{U}}\right) \\ (\frac{a}{n})\left(\tilde{\mathbf{V}}_P\Delta t - \tilde{\mathbf{U}}\right) \end{bmatrix} \tag{9.16}$$

Each mode is independent within this system of equations, and so we may write the Kalman filter for each of the separate modes:

$$\begin{bmatrix} \dot{\tilde{u}}_i \\ \ddot{\tilde{u}}_i \end{bmatrix} = \begin{bmatrix} \dot{\tilde{u}}_i + (\frac{2a}{n})^{1/2}(\tilde{v}_{P,i}\Delta t - \tilde{u}_i) \\ (\frac{a}{n})(\tilde{v}_{P,i}\Delta t - \tilde{u}_i) \end{bmatrix} \tag{9.17}$$

where $\tilde{v}_{P,i}$ is the i^{th} element of $\tilde{\mathbf{V}}_P$.

Having determined the optimal observer equations for mode amplitude and velocity, we can now discretize over time and formulate the displacement prediction for time $t + \Delta t$. For mode i this is

$$\tilde{u}_i^{t+\Delta t} = \tilde{u}_i^t + d_1\dot{\tilde{u}}_i^t + d_2\left(\tilde{r}_i - \tilde{u}_i^t\right) \tag{9.18}$$

which is exactly the central-difference update rule for direct time integration of the finite element governing equations [24, 20], with "loads" $\tilde{r}_i = \tilde{v}_{P,i}\Delta t$, $d_1 = \Delta t$ and $d_2 = 2\Delta t^2/\tilde{m}_i = (a/n)\Delta t^2 + (2a/n)^{1/2}\Delta t$.

The equivalence between these Kalman filter equations and time-integration of a finite-element governing equation provides an intuitive interpretation of the Kalman filter. In essence, it is smoothing the optical flow data over space by modeling it using the low-frequency, whole-

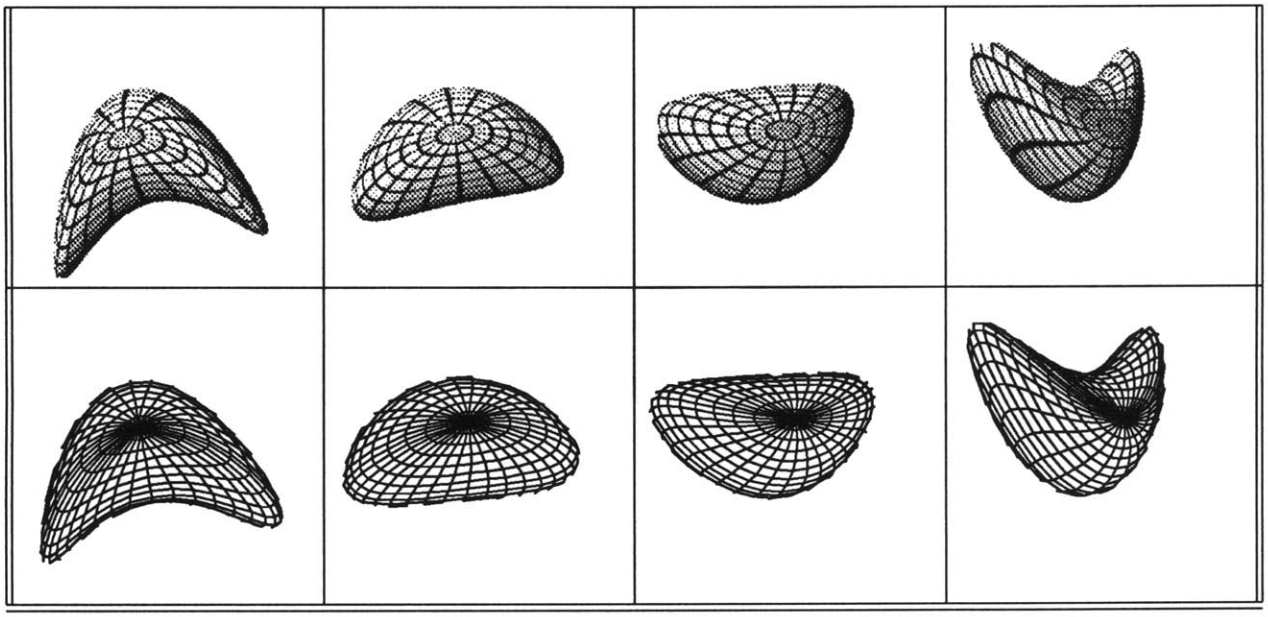

Figure 9-1: Using the Kalman filter to track rigid and non-rigid motion. Top row: Input image sequence, Bottom row: estimated position and shape.

body mode shapes, and smoothing over time by use of a mass matrix **M**,

$$\mathbf{M} = \frac{2\Delta t}{(a/n)\Delta t + (2a/n)^{1/2}}\mathbf{I} \tag{9.19}$$

The effect of the mass matrix is to integrate information across time, thus providing a more accurate estimate than is possible from a single measurement of velocity. When the observation noise is large relative to the acceleration or excitation noise, the solution becomes similar to simple time averaging. When the acceleration noise is large relative to the observation noise, the solution become similar to the single-measurement case.

9.1 An Example Using Synthetic Data

In the kinematic case, our major concern was the behavior of the estimator with increasing levels of noise. In the dynamic case, our principal concern is the convergence and possible bias of the Kalman filter. There is some reason for such concern, as both rotational dynamics and non-rigid dynamics are non-linear problems that are linearized by the FEM. As a consequence, the Kalman filter developed here may be more properly considered an *extended* Kalman filter and, despite the well-known stability and accuracy of the FEM, there is no proof of convergence and bias-free behavior. We therefore have evaluated the stability and accuracy of our formulation using synthetic data.

9.1.1 An Illustrative Example

Figure 9-1 shows an example of tracking both rigid and non-rigid motion. The top row of Figure 9-1 shows four frames from a 30 frame image sequence of a rotating, translating, and deforming solid. In this example the initial position, shape, and (linearized) velocity of the object was assumed known. Exact optical flow data from this sequence was corrupted by uniform noise to produce a 16 dB SNR (approximately 15% noise). This noisy optical flow was used as input to the Kalman filter of Equation 9.18, thus producing estimates of position, shape and motion. Both rigid-body and non-rigid motions were estimated simultaneously. The inter-frame time step was 0.1 seconds, and the parameters of the Kalman noise model were $(a/n) = 0.3$ (large accelerations are assumed to be relatively rare).

The bottom row of Figure 9-1 shows the resulting estimates of shape and motion. In this example the rigid-body modes were tracked with an error of 29.1 dB SNR (approximately 3.5% error). The error in both rigid-body and non-rigid modes was 18.5 dB SNR (approximately 11.9% error). The fact that object motion could be tracked with much less error than was present in the optical flow is attributable to the integration of information across the whole body and across time.

The major source of error in the non-rigid modes was introduced by a single ill-conditioned mode, i.e., a mode for which even large deformations cause small only 2-D motions. In Figure 9-1, for instance, even though the amount of bending perpendicular to the image plane is almost 5% in error the tracking object appears nearly identical to the original object. Such ill-conditioned modes can be detected by examination of the columns of $\mathbf{\Phi}_P$, although this was not done in this experiment.

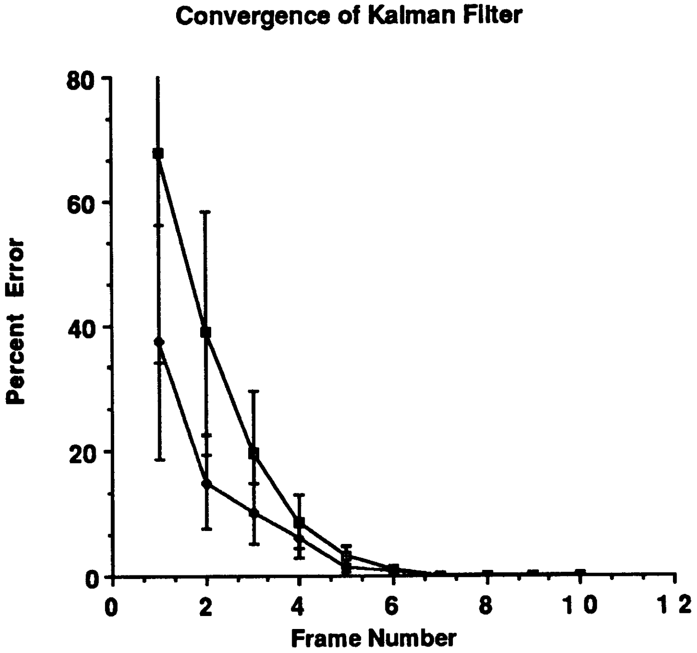

Figure 9-2: Estimation error as a function of frame number, for two ratios of acceleration noise to optical flow field noise. Optical flow SNR is 10 dB.

9.1.2 A Statistical Evaluation

The previous examples have shown that our formulation can produce accurate estimates of motion, but they do not allow evaluation of either convergence or stability. To evaluate these properties, we followed the methodology of Brodia and Chellappa [6], and constructed an experiment in which there were large errors in the initial velocity estimates. This condition is equivalent to the case in which a very large acceleration "spike" produces a large inter-frame change in system velocities. Following this acceleration spike, the behavior of the Kalman filter over successive frames was observed to determine whether or not the Kalman estimates would converge rapidly to the correct value.

In our this experiment noisy motion estimates for 100 image sequences were used as input to Equation 9.18. The motion estimate noise level averaged 20 dB (i.e., the noise magnitude was 10% of the flow vector magnitude). The mean velocity for each mode (including rigid-body modes) was approximately 5 cm/second. The inter-frame time interval was 0.25 seconds. In each trial the initial estimate of each mode's velocity was zero so that the mean initial error was 5 cm/second for each mode. This condition is equivalent to applying an acceleration of 20 cm/second^2 to a resting system between the 0^{th} and 1^{st} frames of the image sequence.

The Kalman filter's output was then observed over the next twelve image frames, as shown in Figure 9-2. Both rigid-body and non-rigid motions were estimated simultaneously. In this figure percent error was measured as

$$\text{Percent Error} = 100 * \left(\frac{\sum \|\hat{\tilde{\mathbf{U}}} - \tilde{\mathbf{U}}\|}{\sum \|\tilde{\mathbf{U}}\|} \right) \tag{9.20}$$

where $\hat{\tilde{\mathbf{U}}}$ is the estimated motion, $\tilde{\mathbf{U}}$ the true motion, and the sums are taken over the 100 experimental trials.

The experiment was repeated with two separate measurement/acceleration noise models, one with $a/n = 2$ (large accelerations are common) and one with $a/n = 0.5$ (large accelerations are uncommon). The upper curve in Figure 9-2 shows the estimates convergence with the $a/n = 0.5$ model, the lower curve shows convergence with $a/n = 2.0$. The error bars show the standard deviation of the 100 separate estimates at each frame number. Note that although the mean error goes very nearly to zero, in individual trials the errors were in the range of ±1%.

As can be seen from Figure 9-2, stable and accurate convergence was achieved in both cases. All modes, including rigid-body modes, behaved in a very similar manner. When the noise model was more appropriate to the large initial acceleration (the case where $a/n = 2.0$) convergence to approximately 10% error was achieved by frame 3. When the noise model was less appropriate (the case where $a/n = 0.5$) convergence to 10% error required 4 frames. In both cases, and for all modes, convergence was both stable and unbiased.

As in the kinematic estimation case, the major factor that permits the stability and noise-resistance shown here is the fact that data is integrated over the entire body rather than only over a small patch. As with the kinematic case, it must be noted that because $\mathbf{\Phi}$ linearizes object motion, it is important that inter-image rotations and deformations remain small.

Chapter 10

Examples Using Real Data

10.1 Examples Using Real Data

Figure 10-1 shows an example of recovering non-rigid motion from optical flow data and Equation 9.18. The upper image in each box shows six successive frames of transmission X-ray data, from which the rigid and non-rigid motions of the heart ventricle were estimated. Time increases from top left to bottom righ; total elapsed time is approximately one second. Optical flow was computed by use of a block-wise version of the Horn-Shunck optical flow algorithm.

The 3-D shape and motion of the heart ventricle was tracked over time using this optical flow data. The computation started with an initial 3-D model of the ventricle, shown in wireframe at the top left of Figure 10-1 (the bottom image in frame 1). See Pentland and Sclaroff [23] for details on obtaining the initial 3-D object description. Equation 9.18 was used to estimate the 3-D rigid and non-rigid motion of the ventricle at each time step. The resulting rigid and non-rigid motions are shown by the wireframe illustrations at the bottom of each frame, overlayed on the original X-ray imagery. Execution time was approximately one second per frame on a standard Sun 4/330.

It can be seen that the 3-D shape of the ventricle model is quite similar to the shape seen in the original imagery. The major defect appears near the top of the wireframe model in frames 3 and 4. Close examination showed that the tracking process became confused in this area because of the large deformations occurring between frames 2 and 3; as a consequence the top edge of the model became "stuck" on edges in the surrounding volume.

Note that the estimated ventricle shape is very nearly the same in the first and last frames, even though there was no constraint maintaining the position, rotation or volume of the model during the estimation process. The fact that the 3-D model returned to its original shape, position, and volume at the point at which the real ventricle returned to its original shape and position is evidence of the stability of the Kalman filter solution.

10.1.1 Constrained motion

In many cases the observed motion is known to be constrained, for example, by gravity or by a hinge or other attachment. Such constrainted motion adds a *bias* or *control* term to Equation 9.1, but as long as it varies sufficiently slowly with respect to the Kalman filter's sampling rate it does not otherwise affect the convergence or stability of the estimator [2, 9]. We may therefore hope to use the Kalman filter of Equation 9.18 to track the rigid and non-rigid behavior of *constrained* objects as well free-moving objects.

In the ThingWorld modeling and simulation system [24, 25, 20, 21], which provides the software base for the work described here, both gravity and spring-like constraints may be used to affect object's behavior. To attach two objects to each other, for instance, a spring constraint is placed between a point on each object's surface, and this constraint exerts equal and opposite attractive forces on the two points of attachment. In the Kalman state equations such forces appear as a constant or slowly-varying acceleration bias, i.e., Equation 9.6 becomes

$$\begin{aligned}\dot{\tilde{\mathbf{U}}} &= \tilde{\mathbf{V}} \\ \dot{\tilde{\mathbf{V}}} &= \mathbf{R}^c + \mathrm{a}\end{aligned} \tag{10.1}$$

where $\mathbf{R}^c$ is a vector describing the load exerted on each nodal point by all active constraints (see reference [21] for additional details). The spring force is proportional to the square of the distance between the two constrained points.

Given a *priori* knowledge of such a motion constraint, we can compensate for the contribution of that constraint to the state equations and then estimate motion as previously. The simplest way to accomplish this is to modify Equation 9.18 to account for this new term:

$$\tilde{u}_i^{t+1} = \tilde{u}_i^t + d_1 \dot{\tilde{u}}_i^t + d_2 \left(\tilde{r}_i - \tilde{u}_i^t \right) + 2\Delta t^2 \tilde{r}_i^c / \tilde{m}_i \tag{10.2}$$

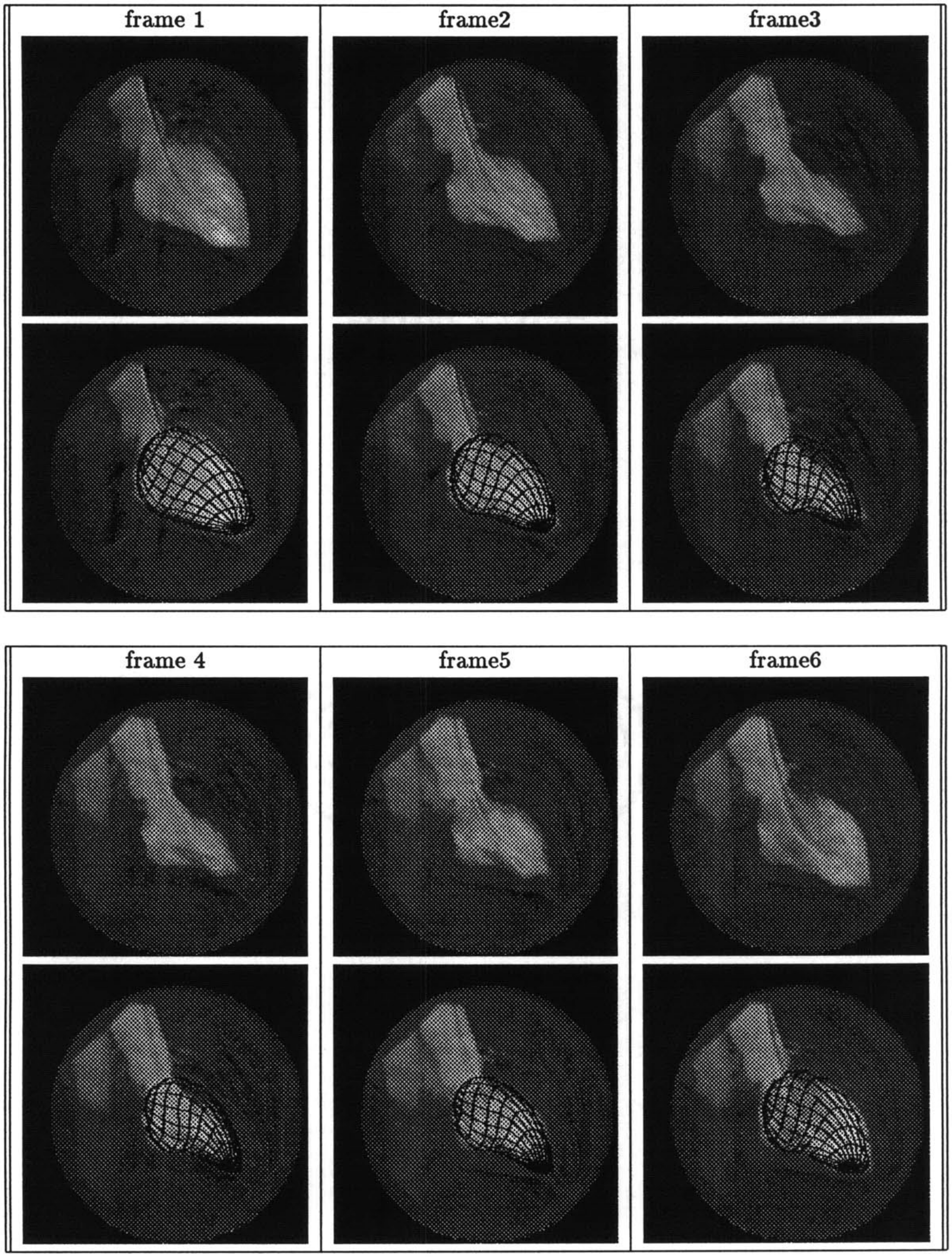

Figure 10-1: Recovery of the rigid and non-rigid motion of a 3-D model of a human heart ventricle; time proceeds from top left to bottom right.

where $\tilde{m}_i = \rho \sum_j |\phi_{i,j}|$ is an estimate of the i^{th} mode's generalized mass, parameterized by ρ, an estimate of the object's density.

Figure 10-2 illustrates a relatively complex example of tracking an object in which motion is a *priori* known to be constrained to certain part junctions (joints). This figure shows three frames from a twelve image sequence of a well-known tin woodsman caught in the act of jumping. Despite the limited range of motion, this example is a difficult one because of the poor quality optical flow, due to pronounced highlights on thighs and other parts of the body.

In this example the initial 3-D model was constructed by hand, with the spring-like constraints described previously inserted between the various body parts. In this manner the combined behavior of the various parts were constrained to be consistent with the articulation of the human body. An example of automatic recovery of a similarly complex 3-D model is shown in Pentland and Sclaroff [23].

Optical flow estimates were then calculated by use of a block-wise Horn-Schunk algorithm, and Equation 10.2 was used to estimate the constrained rigid-body motions of the various parts. In this case only the six non-rigid modes were employed because of the large amount of noise in the optical flow data. The estimates of constrained motion for this sequence are illustrated by the bottom row of Figure 10-2.

As can be seen by comparing the 3-D motion of the model with that in the original image, the resulting tracking is reasonably accurate. To a substantial extent this accuracy is attributable the articulation constraints, as without them errors due to moving surface highlights would have caused the "thigh" parts to fly off in wildly incorrect directions. The inter-part connectivity enforced by these constraints allowed the stable motion estimates for the body and lower legs to counterbalance these erroneous motion estimates.

10.2 Conclusion

We have introduced a precise, physically-correct model of elastic non-rigid motion. This model is based on the finite element method, but decouples the degrees of freedom by breaking down object motion into rigid and non-rigid *vibration* or *deformation modes*. Because of the intrinsic elastic properties of real materials, it can be shown that the high-frequency modes in this representation rarely have significant amplitude, so that they may be discarded without introducing

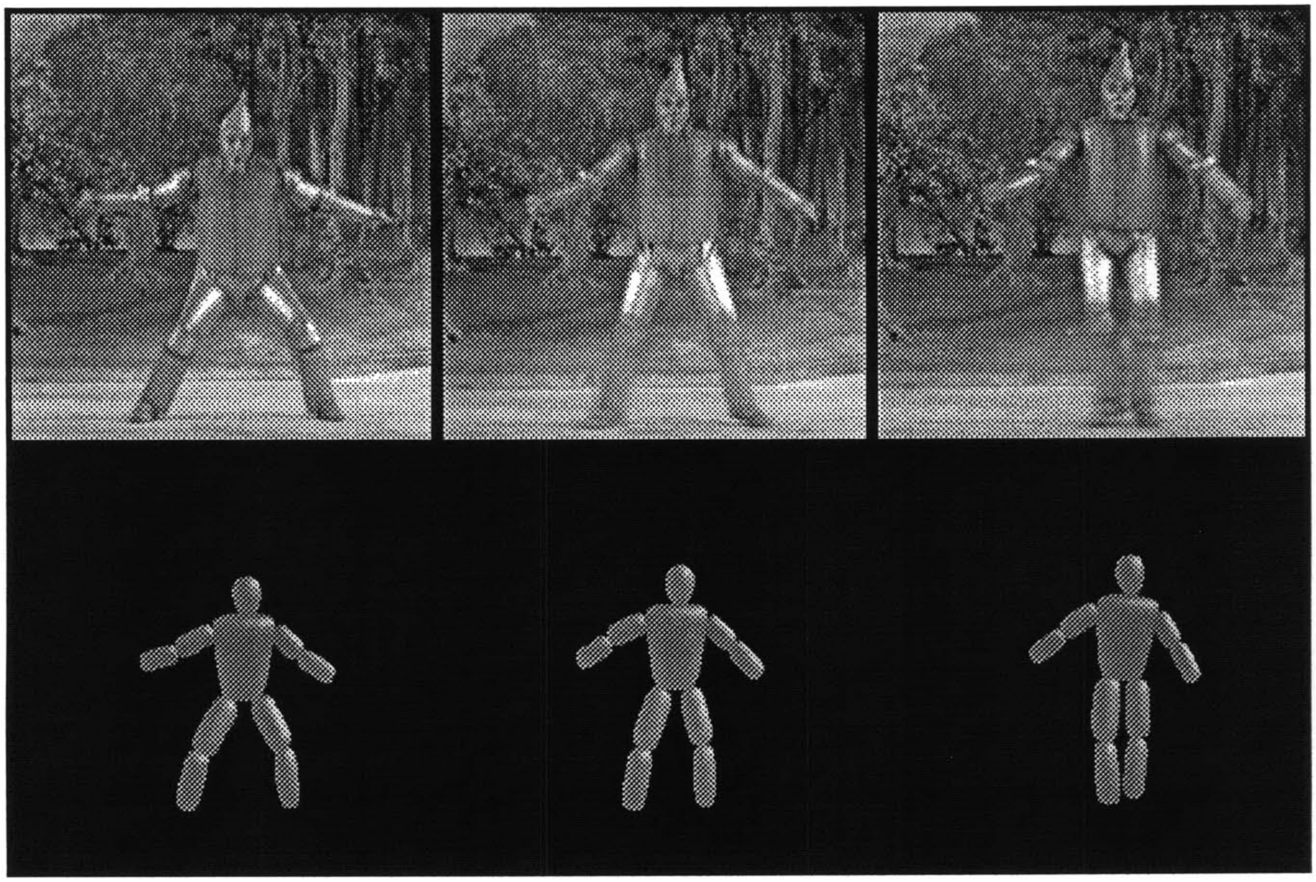

Figure 10-2: Three frames from an image sequence showing tracking of a jumping man using an articulated, physically-based model. Despite poor quality optical flow (due to pronounced highlights on thighs and other parts of the body) the overall tracking is reasonably accurate. This accuracy is in part due to the presence of articulation constraints between the various parts of the model.

undue error.

The result is an accurate representation for both rigid and non-rigid motion that has greatly reduced dimensionality, capturing the intuition that non-rigid motion is normally coherent and not chaotic. Because of the small number of parameters involved, we have been able to use this representation to obtain accurate, overconstrained estimates of both rigid and non-rigid motion.

We have also shown that these estimates can be integrated over time by use of an extended Kalman filter, resulting in stable and accurate estimate of both 3-D shape and 3-D velocity. The formulation was then extended to include constrained non-rigid motion. Examples of tracking single non-rigid objects and multiple constrained objects were presented.

An inevitable limitation of our technique stems from the fact that certain rigid-body and non-rigid motions cannot be observed under orthographic projection. The inability to observe motions such as translation in z, shear along the z-axis, etc., means that errors in estimation of these motions are unavoidable. Further, the situatation is exacerbated when observing extremely simple objects, such as planes or rods, as in these cases there is not enough data to distinguish between various of the modal deformations.

For instance, when observing a rod, rigid-body rotation cannot be distinguished from lengthwise contraction. Note, however, that if the stiffness of each mode is included in the calculation (by scaling each column of $\mathbf{\Phi}$ proportional to the corresponding eigenvalue) then most of the observed 2-D motions will be accounted for by low-frequency modes such as rotation, and relatively little allocated to higher-frequency modes such as non-rigid contraction. The preferential allocation of observed 2-D motion to the lower-frequency modes (including the rigid-body modes) is similar to human observers' well-known bias towards the simplest motion interpretation.

Another limitation of our current technique stems from the use of optical flow, rather than feature points, as the input data. The use of optical flow data requires us to integrate object motion over time in order to determine the object's current position and shape; there is no way to "anchor" our estimates of position and shape to our current observations. As a result small biases in estimating rotation, etc., can grow over time and eventually destroy our ability to accurately track the object. We are therefore working on integrating feature point information into our motion and shape estimates, so that we no longer have to rely completely on time integration to determine the object's current state.

Chapter 11

Conclusion

Two techniques were presented, representative of syntactic and semantic image representations respectively. The first addressed an approach to image compression based on the statistical relationship between subbands of the wavelet transform. A sample implementation demonstrated how these relationships might be discovered and exploited in a practical way. The technique is based on a low-level, syntactic analysis of the image signal. The second technique dealt with the recovery of non-rigid motion from optical flow data. A closed form solution was presented which is based on a principled, physically-based method of over-constraining this traditionally ill-posed problem. The output of such a technique is a high-level semantic understanding of the rigid and non-rigid motion in a scene.

Directions for Further Research

The two systems described herein could be combined to form an integrated image coding and understanding system. The QMF pyramid provides a coding technique which is simple, efficient and allows for real-time transmission of image sequences. Once these images are transmitted, the motion algorithm enables an efficient framework for tracking, as discussed in chapter 10.1.

An implementation of such a system in underway. Images (128x128) are acquired and coded at a rate of about 4 Hz. They are then transmitted to a separate process which tracks the position of the hand using simple low-level processing, and decodes the image. The hand position is then transmitted to the ThingWorld system which tracks the position subject to inertial forces and internal constraints. Real-time performance has already been achieved without special purpose or dedicated hardware.

Bibliography

[1] E. Adelson, E. Simoncelli, and R. Hingorani. Orthogonal Pyramid Transforms for Image Coding. In *Proceedings of SPIE*, October 1987.

[2] M. Aoki. *Optimization of Stochastic Systems: Topics in Discrete-Time Dynamics.* Academic Press, 1989.

[3] Arun Netravali and Barry Haskell. *Digital Pictures: Represantaion and Compression.* Plenum-Press, 1988.

[4] N. Ayache and O. D. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Trans. Robotics and Automation*, 5(6):804–819, 1989.

[5] M. Barnsley. *Fractals Everywhere.* Academic Press, 1988.

[6] T. J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1):90–99, January 1986.

[7] P.J. Burt and E.H. Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Trans. Communications*, COM-31(4):532–540, 1983.

[8] O. D. Faugeras, N. Ayache, and B. Faverjon. Building visual maps by combining noisy stereo measurements. In *IEEE Trans. Robotics and Automation*, San Francisco, CA, April 1986.

[9] Bernard Friedland. *Control System Design.* McGraw-Hill, 1986.

[10] H. Gharavi and A. Tabatabai. Application of Quadrature Mirror Filters to the Coding of Monochrome and Color Images. In *Proceedings ICASSP*, pages 32.8.1–32.8.4, 1987.

[11] J.J. Gibson. *The Senses Considered as Perceptual Systems.* Houghton Mifflin, 1966.

[12] D.J. Heeger. Model for the extraction of image flow. *Journal of the Optical Society of America A*, 4(8):1455–1471, 1987.

[13] K. Bathe. *Finite Element Procedures in Engineering Analysis.* Prentice-Hall, 1982.

[14] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transaction ASME (Journal of Basic Engineering)*, 82D(1):35–45, 1960.

[15] R.E. Kalman and R.S. Bucy. New results in linear filtering and prediction theory. *Transaction ASME (Journal of Basic Engineering)*, 83D(1):95–108, 1961.

[16] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1987.

[17] B. Mandelbrot. *The Fractal Geometry of Nature.* W.H. Freeman & Co., 1982.

[18] A. Pentland. Fractal Based Description of Natural Scenes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 6(6):661–674, 1986.

[19] A. Pentland. Perceptual Organization and Representation of Natural Form. *Artificial Intelligence*, 28(3):293–331, 1986.

[20] A. Pentland. Automatic Extraction of Deformable Part Models. *International Journal of Computer Vision*, pages 107–126, 1990.

[21] A. Pentland, I. Essa, M. Friedmann, B. Horowitz, and S. Sclaroff. The Thingworld Modeling System: Virtual Sculpting by Modal Forces. *Computer Graphics*, 24(2):143–144, 1990.

[22] A. Pentland and B. Horowitz. Recovery of Non-rigid Motion and Structure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13, to appear in July 1991. Special Issue on Physically-Based Modeling.

[23] A. Pentland and S. Sclaroff. Closed-Form Solutions for Physically-Based Shape Modeling and Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13, to appear in July 1991. Special Issue on Physically-Based Modeling.

[24] A. Pentland and J. Williams. Good Vibrations : Modal Dynamics for Graphics and Animation. *Computer Graphics*, 23(4):215–222, 1989.

[25] A. Pentland and J. Williams. The Perception of Non-Rigid Motion: Inference of Material Properties and Force. In *Proc. International Joint Conference on Artificial Intelligence*, August 1989.

[26] A.P. Pentland and B. Horowitz. A Practical Approach to Fractal Based Image Compression. In *IEEE Data Compression Conference*, April 1991.

[27] L. Segerlind. *Applied Finite Element Analysis.* John Wiley and Sons, 1984.

[28] E.P. Simoncelli. Orthogonal Sub-band Image Transforms. Master's thesis, M.I.T., 1988.

[29] M. Subbarro. Interpretation of image flow: A spatio-temporal approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(3):266–278, March 1989.

[30] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-Seeking Models for 3-D Object Reconstruction. In *Proc. First Conference on Computer Vision*, pages 269–276, London, England, December 1987.

[31] Anh Tran, Kwun-Min Liu, Kou-Hu Tzou, and Eileen Vogel. An Efficient Pyramid Image Coding System. In *Proceedings ICASSP*, pages 18.6.1–18.6.4, 1987.

[32] S. Ullman. Maximizing the rigidity: The incremental recovery of 3-d structure from rigid and rubbery motion. *Perception*, 13:255–274, 1984.

[33] M. Vetterli. Multi-dimensional sub-band coding: Some theory and algorithms. *Signal Processing*, 6(2):97–112, 1984.

[34] P. Werkhoven, A. Toet, and J. J. Koenderink. Displacement estimates through adaptive affinities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(7):658–662, July 1990.