Stereo Vision Based on Compressed Feature Correlation and Graph Cut

by

Sheng Sarah Tan

B.S., Precision Instruments, Tsinghua University, 1999
M.S., Mechanical Engineering, Massachusetts Institute of Technology, 2002
M.S., Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2002

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctorate of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2005

Author .....
Department of Mechanical Engineering
), 2005

Certified by .....
Douglas P. Hart
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by .....
Lallit Anand
Chairman, Department Committee on Graduate Students

Stereo Vision Based on Compressed Feature Correlation and Graph Cut

by

Sheng Sarah Tan

Submitted to the Department of Mechanical Engineering
on March 30, 2005, in partial fulfillment of the
requirements for the degree of
Doctorate of Science in Mechanical Engineering

Abstract

This dissertation has developed a fast and robust algorithm to solve the dense correspondence problem with a good performance in untextured regions by merging Sparse Array Correlation from the computational fluids community into graph cut from the computer vision community.

The proposed methodology consists of two independent modules. The first module is named Compressed Feature Correlation which is originated from Particle Image Velocimetry (PIV). The algorithm uses an image compression scheme that retains pixel values in high-intensity gradient areas while eliminating pixels with little correlation information in smooth surface regions resulting in a highly reduced image datasets. In addition, by utilizing an error correlation function, pixel comparisons are made through single integer calculations eliminating time consuming multiplication and floating point arithmetic. Unlike the traditional fixed window sorting scheme, adaptive correlation window positioning is implemented by dynamically placing strong features at the center of each correlation window. A confidence measure is developed to validate correlation outputs. The sparse depth map generated by this ultra-fast Compressed Feature Correlation may either serve as inputs to global methods or be interpolated into dense depth map when object boundaries are clearly defined.

The second module enables a modified graph cut algorithm with an improved energy model that accepts prior information by fixing data energy penalties. The image pixels with known disparity values stabilize and speed up global optimization. As a result less iterations are necessary and sensitivity to parameters is reduced.

An efficient hybrid approach is implemented based on the above two modules. By coupling a simpler and much less expensive algorithm, Compressed Feature Correlation, with a more expensive algorithm, graph cut, the computational expense of the hybrid calculation is one third of performing the entire calculation using the more expensive of the two algorithms, while accuracy and robustness are improved at the same time. Qualitative and quantitative results on both simulated disparities and real stereo images are presented.

Thesis Supervisor: Douglas P. Hart
Title: Professor of Mechanical Engineering

Acknowledgments

It has been a tremendous privilege to work with Professor Douglas P. Hart over the past six years. Many thanks to him for having faith in me when I was fresh out of college, and then at each stage of my intellectual and personal development in this beautiful country; for the freedom he gave me to explore my own ideas, the crucial guidance he offered when needed, the roller coaster trip from lab to consumers that he brought me on, and the uninterrupted generous financial support.

I am deeply indebted to my committee members, Professor Frédo Durand and Professor George Barbastathis. Your constructive and thought-provoking feedbacks on the dissertation are gratefully received.

The influence of Dr. János Rohály on my growth as a researcher cannot be over estimated. I would like to thank him for uncountable hours of discussions on computer vision, optics, coding and other random topics which produced many new ideas.

My fellow officemates Federico Frigerio, Dr. Ryan Jones, Sara Hupp, Karen Davis and Hemanth Prakash have made the lab an enjoyable place to work. Thanks for the many useful group meetings. Also thanks to Sean Buhrmester for his friendship whose help made my graduate life much more bearable.

I would also like to thank my dear friends at Brontes Technologies, Steve Weeks, Joe Boerjes, Tong Zhang, Micah Rosenbloom and Eric Paley for genuinely caring about me all the way along and sharing their expertise.

Lun Li and Dejiao Lin have been good friends and encouragement over the years.

And at the end of this prestigious procession of gratitude are the most important people. Firstly, my parents, who always expected me to be a PhD, provided the moral support to stick at it and allowed me to go so far from home. And finally, my beloved husband, Yaoping, without whose unconditional support and fighter spirit I might have been nobody. Of course, I have not forgotten you guys, Coco and Archie, for bringing joy to my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Designing artificial vision to match the sophistication of human vision has been a passionate dream for computer vision researchers. One possible model for the visual system, either organic or artificial, divides the "seeing" process into two stages as illustrated in Figure 1-1. Two tasks are at hand: acquiring high quality image data and analyzing the data. The second stage of image analysis may be logically divided into three sub-steps.

The human visual system is remarkably fast and robust. The iris automatically dilates and constricts the pupil to allow more or less light into the eye, enabling us to see an intensity range greater than $1,000,000:1$ [38]. Our eyes capture images at a amazing speed and with high fidelity [96]. All is achieved with a simple pair of optical systems and series of detectors, *i.e.*, the retina [18]. Our single-lensed eye has very little built-in facility for aberrations correction [38], which are normally corrected in good quality optical instruments. Yet our perceived images appear perfectly sharp. High quality raw information is important in that it provides a basis for any further brain or post- processing. Over the past decade, optics and digital light detectors have advanced dramatically [53], although the average image quality still cannot match that of the human visual receptors.

Image analysis may be broadly categorized into three steps of processing: pre-processing, low-level post-processing and high-level post-processing. Pre-processing involves various raw image filtering tasks such as sharpening, blurring and noise reduction. This portion of the

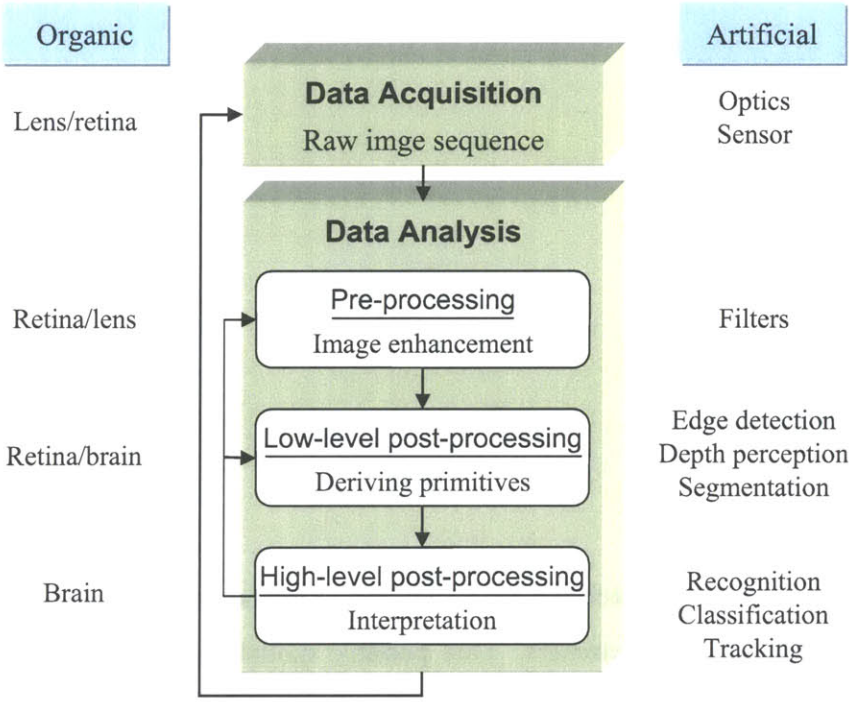

Figure 1-1: The "seeing" process model.

seeing process is bundled with data acquisition for humans because both of them are carried out by the eyeball. Focus of the lens is automatically adjusted by muscles in the eye. Light must penetrate several layers of neural cells on the retina surface before reaching the photo-receptors at the back of the retina. These neural layers are believed to have a function of sharpening contours [90]. In artificial vision, pre-processing is usually carried out after image acquisition due to the mechanical nature of cameras. The second step obtains primitive low-level information. Edge detection and depth perception are sample low-level tasks, which are the subjects of this dissertation. It is known that the optic nerves from the right and left eyes intermix at the optic chiasma on their way to the visual cortex [29]. The intermixing has probably several functions which may include depth perception. Low-level image processing works very fast and allows for instinctive reactions to potential dangers in vertebrates and other groups of animals [118]. In contrast, the last step is a high-level but slower approach, which attempts to interpret the images by compiling them with previous experience and

other sensory information. An example of high-level post processing would be remembering an old acquaintance's face by searching in one's memory. Sample high-level tasks include tracking, recognition and classification. Understanding of the perceived world from the later stages is fed back to affect earlier stages, such as opening up iris or refocusing.

Duplicating the visual capabilities that we take for granted may not seem difficult at first. If only we knew how the human vision carries out its analysis and interpretation, this might lead to a very efficient way of automatic machine vision. However, we are restricted to applying established mathematical theories to artificial vision, because we cannot fully understand the complex mechanism of human vision by probing into the neural networks of the eyes and brain *in vivo* without disturbance, due to the limitation of current bio-engineering technologies.

Computer vision is mainly concerned with image analysis once pictures are taken with existing optical setups. How to achieve comparable speed and robustness of human visual processing remains an especially challenging issue. This difficulty stems from both processing power of hardware and computational algorithms of software.

The human brain is a complex system with more than a hundred billion neurons that come in different shapes and functions and communicate by means of instant electrochemical reactions [133]. Among the numerous neural assemblies in human (and primate) body, the retina of the vertebrate's eye is about the best understood one. Neurons in a human retina are able to perform a million "edge" and motion detections simultaneously. Their processing speed is equivalent to ten one-million-pixel images per second with pixel level resolution [91].

Studies show that computer vision programs may take about a hundred computer instructions to derive a single edge or motion detection from comparable images [91]. Using these numbers, to match the capabilities of the retina may require a computer with a computational power of at least 1,000 Million Instructions Per Second (MIPS). Intel's Pentium runs at about 2,400 MIPS for a 900MHz CPU and the predicted MIPS number for a 4GHz CPU is 20,000 MIPS [143]. The CPU time is further divided among operating system, image acquisition and processing. It is fair to conclude that state of the art artificial vision

technologies can match the speed of the retina on low-level tasks.

However, human vision does not stop at the retina. There is a visual center in the brain to adjust the eyes in real-time and handle high-level tasks. This is where computers fall short of. If we were to measure the caliber of human brain by only counting the number of neurons as the computation capability, the brain is capable of thousands of millions MIPS. In comparison, the most powerful experimental supercomputer nowadays, such as the Deep Blue, is only capable of performing a few million MIPS [58]. Moreover, studies have shown that different people with various levels of cognitive experience may present very different amount of neural synapse connectivity, providing more brainpower by having better connections. In a word, human vision is arguably much faster than artificial ones.

In another aspect, the human visual system is amazingly reliable and flexible. In comparison, the robustness of artificial vision so far heavily relies on controlled experimental parameters, including contrast, scene luminance, object shape, image quality, random noise, local image motion and discrete image sampling rate. In order to develop a versatile computer vision system, we need to know how much blur, histogram equalization, glare reduction, or refocusing should be applied to pre-process any input images like the human eyes do, which remains a work in progress for artificial vision. A truly versatile vision system that may match its human counterpart is one capable of fast optimizing the parameters of image acquisition and three stages of processing until a reliable judgment can be made based on feedbacks. Artificial intelligence or machine learning [22] is a potential candidate to help achieve human vision flexibility by choosing the optimum parameters such as field of view, blurring and level of exposure.

Vision is a learned art. For example, a child may bump his head on a glass wall the first time he encounters one, but probably not the following times. Artificial vision should be able to draw information from previous experience such as recognizing objects using a shape database. State of the art computers might handle well image capturing and some low-level tasks real-time using preset parameters on a specific image type. However, their computing power is not yet capable to achieve video-rate processing when artificial intelligence is

combined with heuristics to resolve various scenes.

In summary, no existing artificial vision system is comparable to its human counterpart in terms of complexity and speed, which makes computer vision an exciting yet still young field.

1.1 Projection

This section reviews the image formation mechanism. Let us consider a world coordinate system whose origin sits at the optical center of the camera as illustrated in Figure 1-2. The XY-plane is parallel to the image plane. The Z-axis lies along the optical axis and points to the image plane. By using this convention, we have a convenient right-hand coordinate system. The photo-detector is positioned at the image plane outside of the camera in Figure 1-2 rather than at the back of the camera for the purpose of viewing clarity.

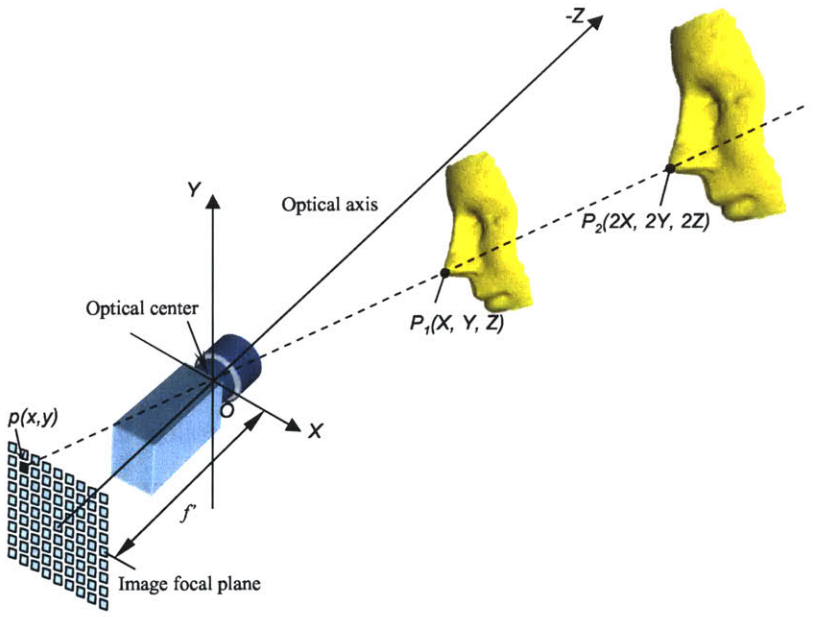

Figure 1-2: Perspective projection.

A simple approximation of the relationship between an object point $P_1(X, Y, Z)$ and its corresponding image point $p(x, y)$ is given by perspective projections [119]:

$$\frac{x}{f'} = \frac{X}{Z}, \text{ and } \frac{y}{f'} = \frac{Y}{Z}. \tag{1.1}$$

where f' is the camera's focal length. Z is always negative in our coordinate system. As a result, x and X have different signs, as well as y and Y. For a more sophisticated imaging model that addresses camera distortions please refer to [37].

There is irreversible information loss when the three-dimensional point P_1 is transformed to a two-dimensional point p. Every point along line OP_1 in the object space is projected to the same point p in the image space based on Equation 1.1. For example, the location of image point p does not uniquely define whether the original object point is $P_1(X, Y, Z)$ or $P_2(2X, 2Y, 2Z)$. Given a single image, we cannot determine whether there is a small object in close range or a large object more further away based on the projection model.

However, humans are able to reliably recover depth information from only one image by covering up one of the eyes. Many high-level visual clues and past knowledge are processed in the brain to accomplish this seemingly easy task, such as object recognition, contour detection, shading and perspectives. For instance, a person standing far away should look smaller. A tilted surface should look darker than a frontal one under head-on point lighting condition. A variety of techniques have been developed with some success based on these high-level cues [65, 83, 127]. For example, in carefully controlled lighting environment, shape from shading is a popular approach to recover 3D shapes from a single image using surface reflectance [2, 74]. If the object's shape is known, such as a bottle or an architecture model, its 3D orientation and location can be reliably determined from one image [85, 104]. However, a single image usually is not enough to determine depth when essential calibration information is unknown such as lighting parameters or rough object shape. A pair, or a sequence of images separated by a known camera displacement, allow depth estimation by triangulation.

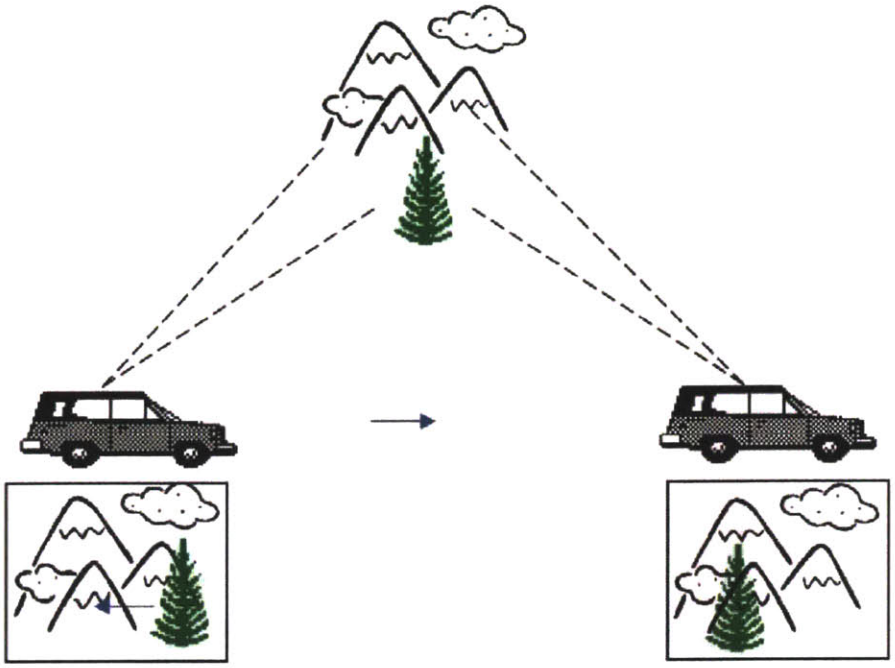

Figure 1-3: Parallax effect.

1.2 Triangulation

Intuitively, triangulation may be explained by the well known parallax effects due to sideways movement. Suppose the object space is composed of a tree in the foreground and a mountain in the far background as illustrated in Figure 1-3. Two images are taken from two different viewpoints. The objects undergo a displacement in the two images. The amount of disparity is inversely proportional to the distance. For instance, the mountain is so far away that it appears in the exact same location of the two images.

Figure 1-4 illustrates a simple camera setup for one-dimensional depth measurement using triangulation. The amount of camera displacement between exposures is called *baseline* b. Assume the optical axises of two cameras are parallel to each other. The image planes and x-axises are perpendicular to the optical axis. The image coordinates in the left and right image, x_L and x_R, are relative to their respective optical centers. The object space origin is set at the middle of two optical centers. z-axis is parallel to the optical axis and points towards the image plane. By comparing two sets of similar right triangles, we have

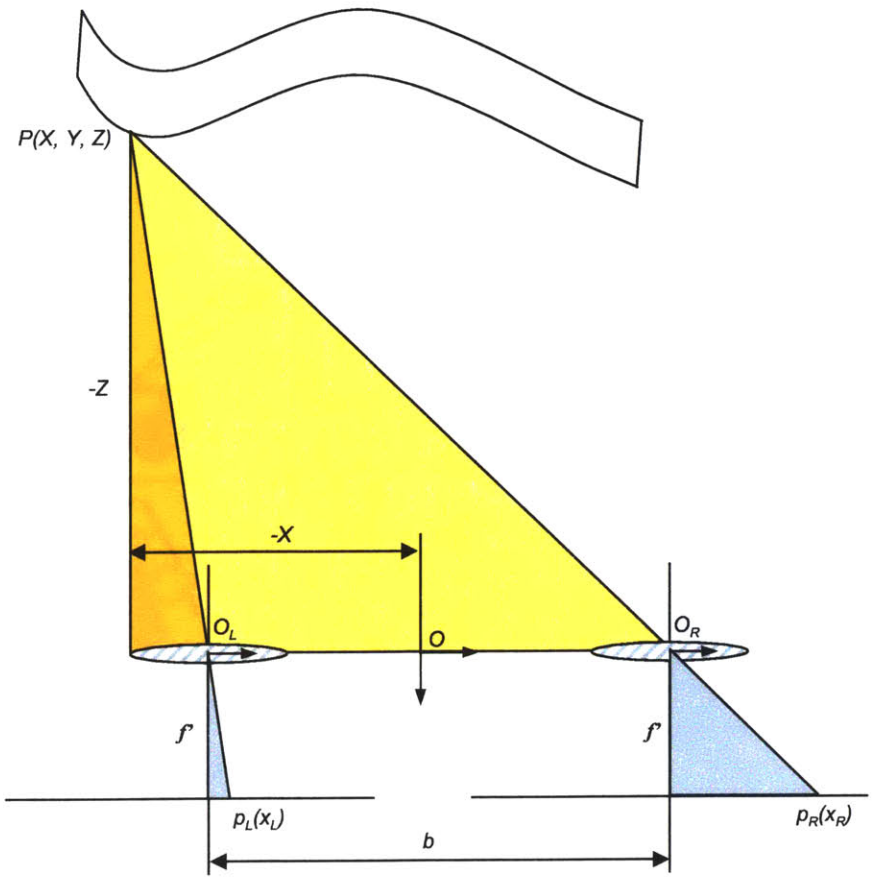

Figure 1-4: Simple camera geometry for triangulation.

$$\frac{x_L}{f'} = \frac{-X - b/2}{-Z}, \text{ and } \frac{x_R}{f'} = \frac{-X + b/2}{-Z}. \tag{1.2}$$

Solving two equations for two unknowns X and Z [56], we get

$$X = b\frac{x_L + x_R}{2(x_L - x_R)}, \text{ and } Z = b\frac{f'}{x_L - x_R}. \tag{1.3}$$

Similarly we can calculate the Y coordinate. The image shifting between frames, $x_L - x_R$, is called the *disparity* d. From Equation 1.3, depth is uniquely determined by disparity and is inversely proportional to d.

Thus, measuring object depth is reduced to the problem of finding image disparity. Calculating the disparity value becomes trivial once we find the corresponding pixels in two

images. Pixel correspondence is the fundamental issue of 3D vision in triangulation-based approaches. In a sense, a depth map directly relates to a disparity map. These two terms are interchanged freely in this dissertation.

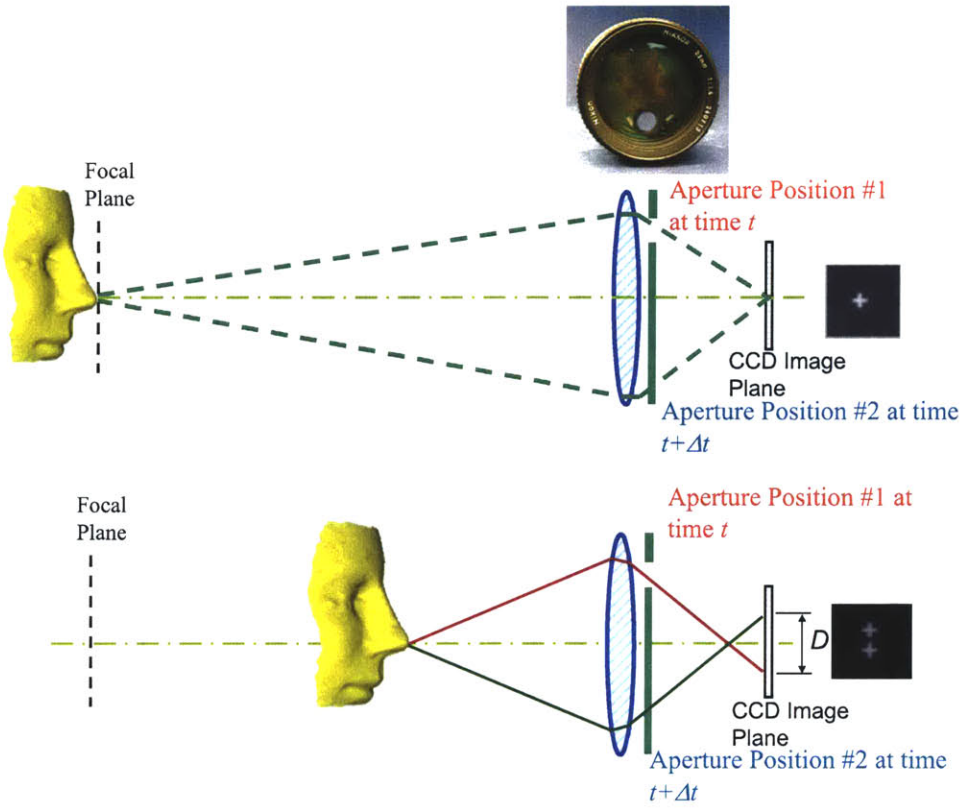

Figure 1-5: Schematics of a single 3D camera with a rotating aperture.

Instead of using two cameras, an alternative hardware setup for depth estimation is to take a sequence of images with a single moving camera. These types of techniques are often called *structure from motion* [24, 43, 44, 93, 100]. Another type of single-camera scheme does not move the entire camera between exposures. Instead, viewpoints are changed by an optical mask inside the camera with two off-axis apertures [10, 75]. An interesting single-aperture variation utilizes an off-axis rotating aperture [105, 106, 121, 128], whose schematics are given in Figure 1-5. Instead of a circular aperture centered on the optical axis as in a standard lens, a motorized disk has one off-axis aperture. As the aperture rotates around

the optical axis, the image point also travels in a circular fashion in the image plane. The diameter D of the circle is reversely proportional to the object point's deviation from the focal plane. Again, depth estimation is reduced to the problem of calculating the disparity between corresponding image points.

1.3 Assumptions and limitations

Figure 1-6: Sample image pair with horizontal disparities.

The Vision problem is especially difficult because in most cases it is under-constrained [78]. Triangulation is a mathematical method to uniquely resolve an object point's 3D coordinates provided that the two corresponding image points can be reliably identified, which is not a trivial task. In an image pair with little intensity variation but significant sensor noise such as the one in Figure 1-6, one pixel p_L in the first image may correspond to many pixels in the other image based on single-pixel intensity matching, when no other assumptions such as smoothness and uniqueness are considered.

Let us suppose an extreme scenario where the object space is consisted of numerous bees distributed randomly. The image of each bee falls on a single pixel on the sensor plane. When two snapshots are taken at two different view points, we have two images with numerous random dots. It is impossible to find each pair of corresponding dots that comes from the same bee. An comparison between computer vision and a physics field might help us understand the complexity of vision problem. In material science, metal or fluids are

comprised of molecules. All neighboring molecules are governed by the same physical laws. As a result, the material basically exhibits homogeneous properties, which enables reliable microscopic measurement of a flow field or cantilever beam. While in computer vision, each image is comprised of thousands of pixels. Each pixel may or may not be connected to its neighbors in the object space depending on whether the pixel's corresponding object point is sitting on the object boundary or not. In our bee example, each pixel corresponds to a different object. Consequently, vision problem is extremely heterogeneous.

Fortunately, computer vision deals with objects larger than bees most of the time. There are a number of assumptions commonly made when solving the correspondence problem to make it more tractable. The assumptions used in this dissertation are discussed below. Each of them has limitations and is not accurate in all cases.

First and most important is the smoothness or continuity assumption. A patch of neighboring image pixels are likely to be formed from the same object of a finite size. Consequently, "disparity varies smoothly almost anywhere" [89]. Smoothness of a surface patch may be modeled as either constant or continuous disparities, while continuity can be further divided into discrete or real numbers [80]. The smoothness assumption usually works well in surface interiors. However, a natural scene may comprise any number and any kinds of objects. The smoothness assumption tends to blur object boundaries, or depth discontinuities.

Second is the uniqueness assumption, which states that each pixel from each image may be assigned at most one disparity value [89]. An one-way uniqueness is implemented in this dissertation, which means that each location (pixel) in the reference image is assigned exactly one disparity value. More sophisticated models enforces two-way uniqueness, which accounts for occlusions by preventing the disparities at multiple locations in one image point to the same location in the other image [3, 35, 60, 70, 80, 142, 146]. The uniqueness assumption implicitly assumes that objects are opaque. For example, the depth of a tree sitting outside a window will probably be detected by triangulation, rather than the transparent window glass or both. Only a few 3D algorithms attempt to address the challenging transparency problem [123].

Third assumption is a Lambertian reflectance model of the world. An ideal Lambertian surface appears equally bright from all viewing angles [56], which means that a given object point should look the same in any camera viewpoints. Color constancy or intensity matching in the case of grayscale images, is the basis for finding pixel correspondence between images. The Lambertian model holds approximately true for most diffusive surfaces. 3D vision algorithms commonly have difficulties with specularities. Camera discretization of intensity values and lighting changes between image frames also invalidate the Lambertian assumption.

The last assumption states that the observed world is stationary in the case of a moving camera. This way, the disparities between images are a pure result of camera viewpoint change, not an effect of object motion. In stereo vision where there are two cameras positioned at different viewpoints, the objects may be dynamic as long as the two cameras are synchronized.

1.4 Three-dimensional vision techniques

Section 1.1 explained that projective imaging using a single camera view is usually not enough for depth reconstruction. Section 1.2 concluded that two or more camera views generally suffice to uniquely determine 3D coordinates based on triangulation. Section 1.3 lists the assumptions that simplify, to some degree, the pixel correspondence problem, as well as their limitations. This section motivates and reviews a few 3D vision algorithms that are related to this dissertation.

Broad adoption of 3D imaging technology is currently limited by speed and robustness. Applications such as robotic surgery [57] and autonomous navigation or tracking [42], demand real-time processing. For example, a texture-mapped 3D view would greatly aid in a surgeon's tactile sense. 3D reconstructed views enable better object recognition without turning the camera and taking more images. 3D object tracking would be much more robust than its 2D counterpart if reliable depth information were available.

There are numerous possible scene types, which may or may not satisfy the assumptions in Section 1.3. Correspondingly there are a large number of 3D algorithms using different

variations of the assumptions [15, 26, 113, 120], which may be successful on certain image types, but so far not all of them. Image types can be broadly categorized in terms of textureness and depth discontinuity.

Finding pixel correspondence is relative easy where there is distinctive feature. How to resolve featureless regions is a classic problem for three-dimensional vision. In the extreme case of a white wall or other uniformly colored surfaces, either white light or laser pattern projection is necessary to add surface texture [19, 120]. 3D methods with special illumination are also called *optical sensing for shape measurement* because of the projective or scanning optics involved. Structured illumination profilometry is a popular way to measure 3D shapes by projecting a known pattern over the surface under test [59]. The projected pattern is then observed by a single camera. The observed pattern is phase-modulated by the topography of the object and depth information is retrieved from the observed pattern using a demodulation process. The projected pattern may take various forms, such as 1D gratings [77, 84, 110, 131, 139, 144], 2D gratings [125, 126], moiré gratings [63, 136], gray- or color-coded gratings [81, 111] and speckles [66, 124, 128]. On the contrary, natural illumination without projected patterns poses a vastly different problem. Inherent surface texture becomes a crucial factor influencing depth estimation accuracy. Many local or global minimization techniques work well with texture rich objects but fail in untextured areas, such as focus/defocus [30, 82, 95] and structure from motion [24, 43, 44, 93, 100].

Whether depth discontinuity is present or the abruptness of discontinuity is another measure for image difficulty. Generally speaking the more abrupt the edges, the harder the image to be resolved three-dimensionally. Due to the complexity of possible image types, not a single 3D algorithm can claim versatility in all vision tasks at the moment. This dissertation addresses images with natural textures and depth discontinuities, but the proposed concept can be easily introduced to other simpler image types with structured illumination and smooth surfaces.

Stereo vision generally refers to the class of applications that detect 3D depth information by comparing images from two different viewpoints [36, 62, 92, 115, 132, 146]. The hardware

requirement for stereo vision is usually simpler and cheaper than multi-camera setup [31, 94, 99], as well as other categories such as laser scanning. This is the reason why stereo vision is one of the most popular methods. Compared with multi-image techniques, stereo vision gains speed by sacrificing the accuracy provided by any additional images. Understanding the state of art in stereo vision ensures a solid foundation for studying multi-image 3D techniques.

There are two general approaches to stereo vision, local methods versus global ones. Local techniques find each pixel's correspondence by computing a cost function in a small interrogation window around the pixel of interest. With the use of smoothness assumption, which implies that neighboring image pixels will likely have similar disparities, information among neighboring pixels are pooled together to reduce ambiguity and false matches. Popular cost functions include SSD (Sum of Squared Differences), cross correlation [51] and optical flow [4, 24, 32, 45, 54, 55]. The biggest advantage of local methods is their capability to obtain a depth estimation for the entire field of view at fast speed. The question is that whether their estimation is accurate and robust enough for all kinds of images, especially in untextured or repetitive patterned areas. Also, averaging in a local window tends to blur any sharp depth discontinuities, where the smoothness assumption fails.

Global energy minimization methods [103, 108] treat the entire reference image as one single window and search for the best depth estimation for the entire field by computing a global cost function. Consequently, computational load of global approaches has dramatically increased compared to local methods due to window size. The advantage is that now ambiguities may be reduced in a spatially smooth global solution. By combining information from a larger area of neighboring pixels, global methods are able to propagate reliable estimations into untextured regions. There is no block effect like those in local methods related to small averaging windows.

1.5 Motivation and contributions

Two objectives of three-dimensional computer vision are high processing speed and robustness. Local window-based methods are able to reach video-rate update speed but more vulnerable to image noises. Global energy minimization methods such as graph cut can handle a variety of image types by optimizing over the entire image field at the price of long processing times. This dissertation presents a method that addresses these challenges and make it possible to speed up graph cut automatically while at least maintaining accuracy.

In the past couple of decades, graph cut has emerged as a powerful global minimization technique in computer vision. It consists of two major components. First an energy model is formulated based on several energy terms, as required by all global energy minimization techniques. Each term measures how bad a possible global solution is. The closer the solution is the lower the energy. Total energy is usually calculated as the sum of several energy terms. The goal is to find the solution with the lowest total energy. The second step is to actually find the minimum energy. Compared with other global energy minimization schemes such as simulated annealing [68, 137] or gradient descent [134], graph cut algorithms converge to local minima very close to the global minima in a relatively short time. Consequently, graph cut is chosen as the global minimization technique in this dissertation for its speed advantage.

Current energy models of graph cut in vision have two types of constraints, or energy terms. The first one is the data energy term which favors solutions that match corresponding pixel intensity values. The second one is the smoothness energy term which favors solutions that encourage spatial continuity.

There is no energy term for any prior knowledge in the standard graph cut energy model. The user may often know the 3D position of some targets in the scenes. For example, a blue sky backdrop has a depth of infinity or a disparity of zero. Graph cut with standard energy model may assign a finite depth to the sky due to image noise from frame to frame. Or, when a 3D model of a patient's body is measured, there are often tracking targets attached to his body whose positions can be separately and reliably monitored by laser or ultrasonic

devices [109]. It is very desirable to integrate any prior 3D information of these target points into global minimization models when solving for a dense 3D map. However, because there is not an energy term accounting for prior information in the standard energy model, the valuable target locations are irrelevant to current graph cut approaches.

At first glance, it would appear that this problem can be solved easily by setting correct initial conditions at the beginning of the energy minimization process, because it is well known that in continuous minimization, a more accurate initial condition often implies faster convergence and higher probability to find the global minimum rather than a local minima. However, graph cut is a combinatorial optimization problem that only computes discrete-valued disparities [21]. Also, the image surface is not continuous but rather consists of discrete pixels. As a result, the initial condition behavior in the continuous domain does not apply in the discrete domain. Graph cut finds a minimum solely based on the energy model and is not sensitive to initial conditions. Even if ground truth is supplied as initial conditions, graph cut is bound to deviate from the perfect initial conditions if image noise is present.

Another disadvantage of standard graph cut is speed. Real-time depth map update is very essential for robotic navigation, 3D surgery or product line inspection. At the moment, the fastest graph cut algorithm takes tens of seconds or even minutes to process two real images of a size around 500×500 *pixel*, or a quarter of a million pixels each image. One practice that slows things up is because several graph cut iterations are necessary for convergence.

A third disadvantage of this global minimization technique is its sensitivity to parameter settings in its energy model. It is desirable that a large range of parameter values is applicable on various images with different contrast or complexity.

The main goal of this dissertation is to apply graph-based methods to the problem of reconstructing 3D disparity maps of naturally illuminated scenes from two images taken at different viewpoints. By addressing the three disadvantages of standard graph cut mentioned above, speed, accuracy and robustness are improved at the same time.

A new energy model for graph cut is introduced in this dissertation, which takes account

of prior depth or disparity knowledge. The priored pixels serve as anchor points to stabilize and speed up the minimization process. In addition, a Compressed Feature Correlation algorithm is proposed to provide reliable priors by generating a sparse depth map, which is very fast compared to graph cut because only high-intensity gradient regions are used in cross-correlation. These two building blocks, Compressed Feature Correlation and priored graph cut, can be independent from each other and be integrated with other techniques. For instance, priors for graph cut can be obtained by any other means, either user intervention or outputs for other 3D positioning systems. Likewise, the output of Compressed Feature Correlation can be useful to other high-level vision tasks such as segmentation and recognition.

1.6 Dissertation outline

The rest of the dissertation is organized as following. In Chapter 2, some background on image compression are given, coarse and fine correlation are described and two designs, adaptive window and confidence measure, for improving Compressed Feature Correlation reliability are explained. In Chapter 3, qualitative and quantitative results using Compressed Feature Correlation on both simulated and real images are presented. How to use the sparse depth maps generated by Compressed Feature Correlation as input to segmentation to obtain dense depth maps is also discussed. Most of the work presented in Chapter 2 and 3 was previously published in [129]. In Chapter 4, some background on energy minimization and graph cut are given, difficulties of the standard energy model are listed and finally a new energy model with priors is proposed. In Chapter 5, experimental results in terms of accuracy and speed using the hybrid approach based on both Compressed Feature Correlation and modified graph cut model are demonstrated. Detailed analysis on the entire approach's limitations are also included. Chapter 6 summarizes contributions of this dissertation and recommends future work.

Chapter 2

Compressed Feature Correlation

In this chapter, a fast and robust Compressed Feature Correlation algorithm based on image compression and cross-correlation is developed for correspondence matching. The output is a sparse disparity map, which will be used as prior information for graph cut as discussed in Chapter 4.

Some background on feature-based computer vision algorithms are given in Section 2.1. Section 2.2 summarizes the major symbols used in Chapter 2 and 3 for easy reference. Section 2.3 proposes a simple and fast approach of image compression that detects and saves strong features into a sparse array. Section 2.4 explains how to correlate two images in a compressed format. Compressed Feature Correlation is introduced into computer vision for its amazing speed. In order to control the output accuracy, two additional designs, adaptive window and confidence measure, are developed and detailed in Section 2.5 and 2.6. Most of the time, coarse correlation is adequate where all pixels in a correlation window are assigned a uniform disparity value. When a higher spatial resolution is desired, fine correlation can be performed based on the results of coarse correlation as described in Section 2.7.

2.1 Introduction

Features such as edges and corners play an important role in human vision. One prominent message from psychophysical experiments is that, at least for solid objects with low textural content, it is the first difference of input luminance which predominantly controls visual performance. The visual cortex is especially responsive to strong features in a scene [116]. Together with related abilities such as correspondence matching and tracking, humans are able to react quickly to the environment and focus attention on objects of interest.

The significance of such features is fully recognized in computer vision [40, 117, 135]. For example, one traditional class of techniques applied to facial recognition is based on the computation of a set of geometrical features from a picture of the face such as the sizes and relative positions of eyes, mouth, nose and chin [16, 33]. There is even belief that edge representations may contain all of the information required for the majority of higher-level tasks [28].

Feature-based 2D tracking is extensively implemented in automated surveillance, robotic manipulation and navigation. Because real-time processing is a necessity in these applications, only perceptually- significant information such as contours is retained from video feeds. If the target's 3D model is known, its detected contours are compared against its geometrical model to determine the object's current position and orientation [27]. If there is no *a priori* knowledge of the target, it is tracked by finding the contours' disparity between frames using cross-correlation [25] or level sets [88].

Passive 3D imaging can be reduced to the problem of resolving disparities between image frames from one or several cameras. Some key issues involved are lack of texture, discontinuity and speed. The numerous algorithms that have been proposed to address these issues fall into three broad categories: feature-based, area-based and volume-based algorithms [113]. Same as in 2D tracking, feature-based 3D imaging techniques are able to process an extensive amount of video data in real time while providing enough latency for high-level tasks such as object recognition [52, 101]. This group of methods generates sparse but accurate depth maps at feature points and excels at determining object boundary positions where

area-based techniques often fail. When a full-field depth map is desirable, the sparse 3D representation provides a solid foundation for additional area- or volume-based algorithms to fill in the voids when there is ample surface texture; otherwise, when texture is scarce or highly repetitive, object segmentation methods and interpolation are preferable [61, 87].

In the emerging field of image-based 3D modeling and editing, which has many applications in architectural design and entertainment, long and tedious human efforts are required to manually extract layers and assign depths to a 2D image [97]. Automatic feature-based depth detection would greatly facilitate this process.

2.1.1 Related work

Current methods of finding feature correspondence can be categorized into global or local techniques. Global approaches to the sparse correspondence problem handle the entire set of sparse points by forming a global optimization function. Various constraints such as color constancy, continuity, uniqueness and epipolar constraints guide the search of a global solution. [5, 86]. Global techniques are usually robust but relatively slow due to the iteration and optimization process.

Local methods find each pixel's correspondence by computing a cost function in a small interrogation window around the pixel of interest. Popular cost functions include SSD (Sum of Squared Differences) and cross correlation. Sparse Array Image Correlation [46] is commonly implemented in the field of Particle Image Velocimetry (PIV), where fluid fields are seeded with fluorescent tracer particles and illuminated with a laser sheet [1]. Flow motion is measured by tracking particle displacement [112, 140, 141]. PIV images are comprised of millions of bright spots over a dark background. Each image is compressed into a subset of pixels before correlation that only include high-gradient areas [48, 49, 50]. This technique is especially fast and robust at handling large data sets. The algorithm presented in this chapter shares the same computational grounds as Sparse Array Image Correlation.

Depth discontinuities have been a major concern in area-based stereo matching. Boundary overreach, where the detected boundary locations deviate from the real boundaries,

often occurs when the interrogation window contains both the boundary and its adjacent smooth surfaces. Adaptive window techniques have been developed to solve this problem [20, 64, 100, 130, 138]. An asymmetrical window is set around the pixel of interest so that the interrogation window does not cover the object boundary. A cost function is calculated for each possible window location around the pixel of interest and the window with optimal result is chosen. The disadvantage of such adaptive window schemes is that computational load is increased by an order of magnitude due to traversing through all the possible windows.

2.1.2 Contribution

Speed and precise recovery of feature locations and disparities are the two main goals of the algorithm presented here. Ultra-fast speed is achieved by image compression. The nonessential information to a reliable correlation output is discarded and only strong features are retained. Edge detection, compression and correlation are carried out at the same time to achieve maximum efficiency, which distinguishes Compressed Feature Correlation from the typical correlation or SSD-based techniques. The remaining pixels are stored in sparse format along with their relative locations encoded into 32-bit words. Compression dramatically increases speed because only a fraction of the original pixels are retained for correlation. By introducing the well-established gradient-based compressed image correlation algorithm from the computational fluids community to the computer vision field, real-time scene reconstruction may gain new momentum.

Coarse correlation is first performed to obtain an integer-pixel disparity estimation using large adaptive interrogation windows. Then fine correlation with smaller windows resolves each on-edge pixel's disparity to sub-pixel resolution based on the rough estimation from coarse correlation. Error correlation is chosen over standard cross-correlation because pixel comparisons are made through simple integer calculations rather than the computationally expensive multiplication and floating point arithmetic. In order to avoid the boundary overreach problem, adaptive window positioning is also utilized. However, in the proposed

algorithm interrogation window selection is integrated with edge detection. The optimum window location is explicitly determined at the moment an edge is detected without the need of testing through a series of possible windows.

In the remaining of this chapter, Section 2.2 presents nomenclature used in Chapter 2 and 3. Section 2.3 describes the intensity gradient compression method and the significance of threshold setting. Section 2.4 explains error correlation in a compressed format. These two sections provide the computational grounds of Compressed Feature Correlation. Section 2.5 shows how the appropriate window location is adaptively selected and its advantages. Section 2.6 introduces a confidence measure to constrain reliable correlation outputs. In Section 2.7, fine correlation combined with depth-based segmentation and interpolation is presented as a possible approach to generate a complete depth map.

2.2 Nomenclature

The major symbols used in Chapter 2 and 3 are listed in Table 2.1 for quick reference.

Symbol	Explanation
Φ	Correlation function
Δ	Correlation search length [*pixel*]
Δj, Δk	Indices difference in pixel image [*pixel*]
∇	Gradient operator
λ	Confidence measure fraction
μ	Confidence measure threshold
C_threshold	Compression threshold [*grayscale*]
I	Pixel intensity [*grayscale*]
j, k	Image coordinates [*pixel*] of the j^{th} column and k^{th} row
m, n	Data array indices [*pixel*]
M, N	Interrogation window width and height [*pixel*]

Table 2.1: Nomenclature used in Chapter 2 and 3.

2.3 Image compression

Cross-correlation is a time-consuming process especially when the correlation window size is large. However, not all pixels contribute equally to disparity estimation. Featureless, or low-gradient regions contain little or even noisy misleading information for correlation. Thus, it is a waste of computing time to perform correlation in such regions. Image compression is an essential way to pick out high-gradient areas for correlation and ignore the others.

This dissertation's first contribution is choosing a proper compression scheme for the correspondence problem in 3D vision. Contrary to popular image compression formats such as JPEG, the goal of compression in correlation is not high fidelity restoration which requires both low and high spatial frequency information, but only to keep high-intensity-gradient areas which determines the correlation accuracy. In other words, image compression for correlation is basically an edge detection technique. There are several popular edge detection schemes such as zero crossing, Laplacian, Sobel, Prewitt, Roberts and Canny methods. They usually require global filtering as pre-processing which takes time. The biggest advantage of Compressed Feature Correlation is speed. A local compression scheme is most preferable so that, as each edge pixel is found, it is correlated while other non-edge pixels are discarded. For strongly bipolar image types, like the ones in Particle Image Velocimetry of fluids analysis, two compression intensity thresholds can be set by the user based on a compression ratio parameter [47]. Any pixel with an intensity value smaller than the lower threshold is discarded as dark background. Likewise, any pixel larger than the upper threshold is treated as peak overflow. This way, only high-gradient regions are kept which occupy the middle range of the intensity histogram. For a general purpose grayscale image, a simple and fast compression scheme is needed to calculate local gradient.

The first step in compressed image correlation is to generate a data array that contains just enough information to determine the disparity between two images. From the statistical cross-correlation function,

$$\Phi_{\Delta j,\Delta k} = \frac{\sum_{m=1}^{M}\sum_{n-1}^{N}[I'_{m+\Delta j,n+\Delta k}\cdot I_{m,n}]}{\sqrt{\sum_{m=1}^{M}\sum_{n=1}^{N} I^2_{m,n}}\cdot\sqrt{\sum_{m=1}^{M}\sum_{n=1}^{N} I'^2_{m+\Delta j,n+\Delta k}}} \tag{2.1}$$

where I and I' denote the two corresponding images of a image pair, it is clear that pixels of low intensity contribute little to the correlation coefficient while pixels with high intensities have a much more significant weight due to squaring. This is the reason why cross-correlation produces spurious vectors when there is a flare in one image caused by environmental lighting fluctuations. Correlation also fails in featureless, low intensity gradient regions where camera noise becomes significant. Much of the sub-pixel accuracy in image disparity comes from the pixels residing on edges. Thus, discarding low intensity areas and taking into account only strong features that are relatively insensitive to noise, improves correlation robustness.

In the Compressed Feature Correlation algorithm presented in this dissertation, local spatial gradients are calculated for each pixel by comparing the intensities of every other pixel instead of two neighboring pixels in order to preserve a wider edge. This parameter can be set to even larger numbers than two. However such large settings may not be a close approximation of local intensity gradients especially in areas with dense features. For gradients in both horizontal and vertical directions, the local gradient is approximated as:

$$\nabla(j,k) \approx |I(j+2,k) - I(j,k)| + |I(j,k+2) - I(j,k)| \tag{2.2}$$

If disparities between two images only occur in a known direction, for example, horizontally, the gradient formula is reduced to:

$$\nabla(j,k) \approx |I(j+2,k) - I(j,k)| \tag{2.3}$$

For the sake of simplicity, this dissertation only deals with the horizontal disparity case. This algorithm is easily applicable to two-dimensional disparities.

When the local gradient is larger than a preset threshold, *e.g.*, $C\_threshold = 20$ *grayscales*, the pixel of interest is retained and saved into a sparse array comprised of 32-bit

long words along with its relative locations. Each 32-bit long word is divided into three sections: the last 8 bits store the pixel intensity, the middle 12 bits the y-index k and the first 12 bits the x-index j. For example, a pixel of intensity $I = 60$ at location $j = 1078$ and $k = 395$ is saved as 00011000101101000011011000111100 binary. Storing data in this compressed format significantly reduces the number of memory calls that must be made during correlation. The values of j, k and I can be quickly retrieved in a couple of CPU clock cycles by bit-shifting which is optimized for speed in most processors.

The above gradient criterion is chosen for its simplicity and small region of support. The major concern here is extracting high-intensity-gradient pixels for correlation, not a complete edge map of enclosed contours. Other popular edge-detectors such as Canny, Sobel and Gaussian are not only computational expensive but also require global filtering before edge detection [56]. This is impossible for sparse array format where a simple block transfer cannot be done as in uncompressed format correlation.

A proper threshold is essential to both speed and robustness. The higher the threshold the faster the algorithm because less pixels are stored in the sparse array for correlation; also better robustness because only major object boundaries are detected and minor image textures are omitted. The overall compression ratio is determined by both the image complexity and *C_threshold*. Figure 2-1 illustrates the threshold's role in extracting strong features. At a lower threshold, not only the object boundaries but also untextured areas such as the wall and table are detected. At a higher threshold only the clean edges are extracted.

2.4 Cross-correlation in compressed format

For each correlation window in the second image, the local gradient is first checked at each pixel in the window and qualified pixels are stored in compressed format. If the total number of pixels retained is small, it is considered an empty region and this window is discarded without correlating. If the number is large, *e.g.*, 9 pixels in a 32×32 *pixel* window, then each pixel in the same correlation window in the first image is immediately compressed and cross-correlated against the saved sparse array of the second image before the program moves

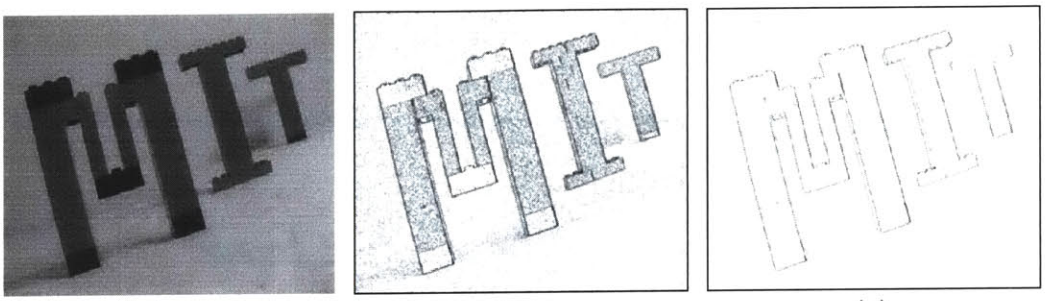

Figure 2-1: Image compression example. (a) The original right image. (b) $C\_threshold = 5$ *grayscales*. Data Retained = 24.1%. (c) $C\_threshold = 15$ *grayscales*. Data Retained = 2.86%.

on to the next window.

This compressed correlation technique is most efficient when there is a minimum amount of overlap among interrogation windows. If there is significant overlap, the number of redundant memory calls and arithmetic calculation from repetitive gradient checking and correlation entries greatly slows processing.

Error correlation is implemented rather than the traditional statistical correlation function because it replaces multiplication with the much faster addition and subtraction. In addition to being faster, it does not place an unduly significant weight on the high-intensity pixels as does the statistical correlation function. It is shown that error correlation significantly improves processing speed while maintaining the level of accuracy compared to the statistical correlation function [107]. The 2D error cross-correlation function can be expressed as:

$$\Phi_{\Delta j,\Delta k} = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}[I_{m,n} + I'_{m+\Delta j,n+\Delta k} - |I_{m,n} - I'_{m+\Delta j,n+\Delta k}|]}{\sum_{m=1}^{M}\sum_{n=1}^{N}[I_{m,n} + I'_{m+\Delta j,n+\Delta k}]} \tag{2.4}$$

or

$$\Phi_{\Delta j,\Delta k} = 1 - \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}[|I_{m,n} - I'_{m+\Delta j,n+\Delta k}|]}{\sum_{m=1}^{M}\sum_{n=1}^{N}[I_{m,n} + I'_{m+\Delta j,n+\Delta k}]} \tag{2.5}$$

The 1D error cross-correlation function in the horizontal direction is simplified to:

$$\Phi_{\Delta j} = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}[I_{m,n} + I'_{m+\Delta j,n} - |I_{m,n} - I'_{m+\Delta j,n}|]}{\sum_{m=1}^{M}\sum_{n=1}^{N}[I_{m,n} + I'_{m+\Delta j,n}]} \tag{2.6}$$

or

$$\Phi_{\Delta j} = 1 - \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}[|I_{m,n} - I'_{m+\Delta j,n}|]}{\sum_{m=1}^{M}\sum_{n=1}^{N}[I_{m,n} + I'_{m+\Delta j,n}]} \tag{2.7}$$

While the typical statistical correlation function computes one entry at a time, error correlation is calculated at the same time as the sparse array is being generated. The entire correlation table is constructed by summing entries as they are found in one interrogation window while traversing through the sparse image array generated from the other corresponding interrogation window. The resulting disparity is obtained by searching for the peak in the correlation coefficient plane. Simple bilinear interpolation is used to determine the correlation maximum within sub-pixel resolution. Compressed error correlation gives a very steep peak, which is ideal for bilinear interpolation.

Sample pseudo-code for simultaneous compression and cross-correlation is presented as following:

```
1 void CoarseCorr(void) {
/* Compress the second image subwindow in the horizontal direction */
2   for each pixel p in the second window
3     calculate local intensity gradient at p
4     if p is a feature point (gradient > C_threshold)
5       save p into sparse array
6     end if
7   end for p
8   if the total number of retained pixels is not trivial in second window
/* Compress the first image subwindow and correlate */
9     for each pixel p in the first window
10      calculate local intensity gradient at p
11      if p is a feature point (gradient > C_threshold)
12        for every q in sparse array
13          compare p and q, and add to correlation table
14        end for q
15      end if
16    end for p
17    find peak position in correlation table
18  end if
19 }
```

2.5 Adaptive window positioning

In the traditional fixed-window position scheme, the entire image is evenly divided into uniformly spaced correlation windows, with or without overlapping. Such a window sorting method is easy to implement. However, it produces more spurious vectors than adaptive window approaches. Gross errors occur when an edge sits across two fixed correlation windows as illustrated by Figure 2-2.

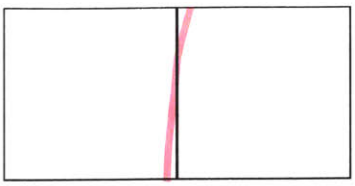

Figure 2-2: Demonstration of a scenario where an edge sits across two fixed neighboring correlation windows.

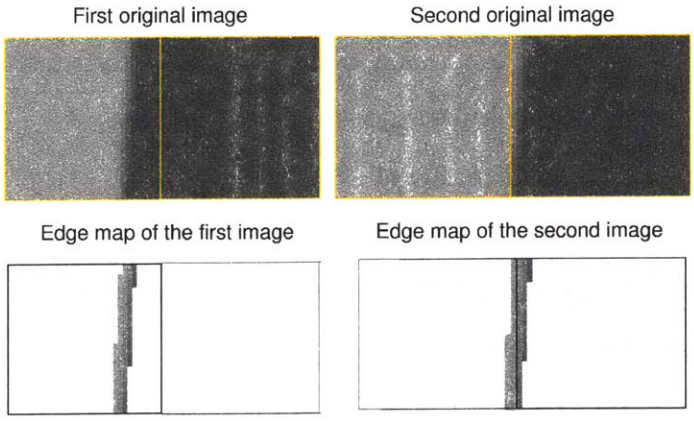

Figure 2-3: Image pair with two fixed neighboring windows cutting the edge in the shifted image. The top shows the original images and the bottom their corresponding extracted edge maps.

Figure 2-3 is a real example. The edge in the first image is shifted by 8 *pixel* to the right. Both the original images and their corresponding extracted edge maps of two neighboring fixed correlation windows are shown. The two windows cut the edge in the second image. As a result, the measured disparity in the left window tends to be smaller than the true disparity because the edge is fully present in the first image and thus has a higher weight in the correlation table. Following the same logic, the right window gives a larger disparity estimate. Correlation result of the left two blocks is 7.23 *pixel*, and the right two blocks 8.88 *pixel*.

In contrast, adaptive window positioning technique enhances robustness by intelligently placing the edges about the center of each correlation window. Each window is dynamically

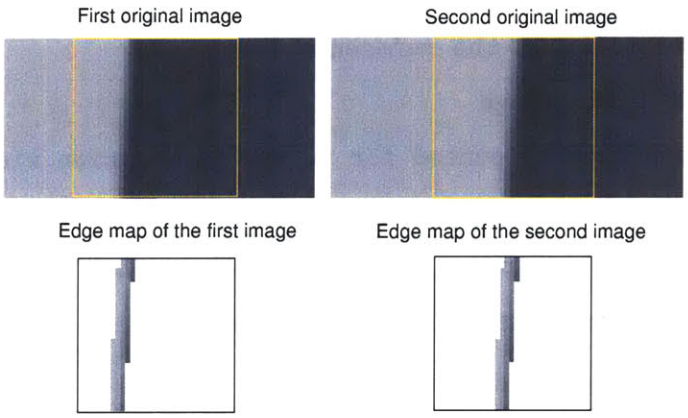

Figure 2-4: Adaptive window positioning is applied to the image pair in Figure 2-3. The top shows the original images and the bottom their corresponding extracted edge maps.

selected at the time an edge is detected. A searching scheme is devised so that when an edge pixel is extracted in the second image, a correlation window is immediately placed around this pixel. All the pixels in this window are now accounted for. The algorithm does no more searching for additional interrogation windows in this block in order to maximize speed and minimize window overlap. Thus, the edge in the images shown in Figure 2-3 is covered by only one correlation window using adaptive window positioning rather than two as with fixed-windows. Figure 2-4 shows the image pair using adaptive window positioning. Now one dynamically positioned correlation window holds the complete edge in both images. Consequently the correlation result of this single window gives the correct 8 *pixel*.

The pseudo-code that demonstrates how correlation windows are adaptively placed is shown as follows:

```
1 void FindBlock(void) {
/* compress horizontally in the reference image.  */
2   for each pixel p in image
3     if p has not been accounted for in any correlation windows
4       calculate local intensity gradient at p
5       if p is a feature point (gradient > C_threshold)
6         check whether its neighboring pixels are also features
7         if yes
8           place a new window here
9           save window location
10          perform coarse correlation:  CoarseCorr()
11         end if
12       end if
13     end if
14     pixel pointer jumps outside this window
15   end for p
16 }
```

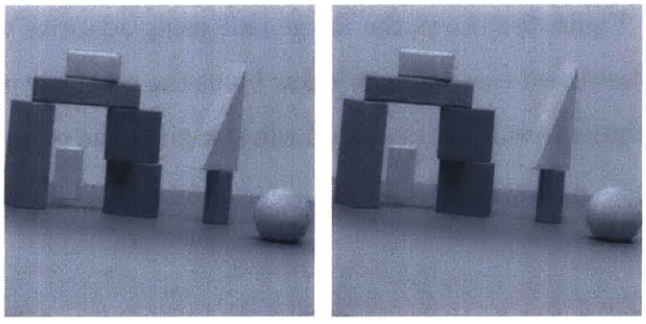

Figure 2-5: Left: the original image of several building blocks. Right: the second image is artificially shifted to the right by 8 *pixel*.

The following example demonstrates the effectives of adaptive windows over fixed ones. The first image in the image pair as illustrated in Figure 2-5 is captured by a camera with

an image size of 500×500 *pixel* [17]. The second image is obtained by simulating a uniform lateral disparity of 8 *pixel* relative to the first image. Integer disparity simulation is simply obtained by pixel index shifting. Compression and correlation are only performed in the X-direction using the proposed edge matching algorithm as there is no vertical shift.

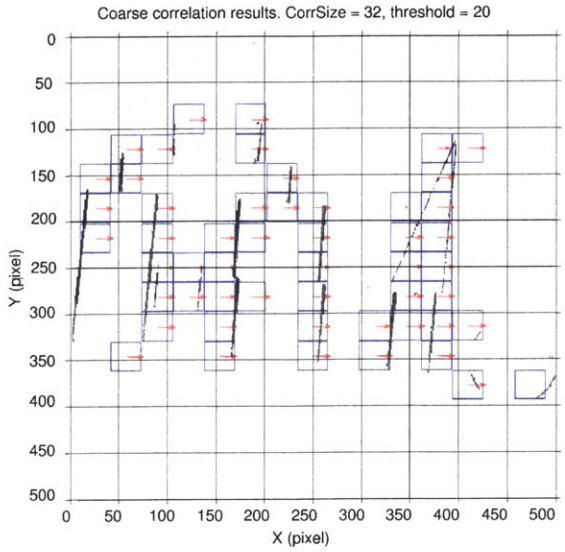

Figure 2-6: Compressed coarse correlation results of an image pair with a simulated horizontal disparity of 8 *pixel* using fixed windows. Each block represents a non-empty correlation window. (Only the edge map of the second image is shown.)

The measured disparity results using fixed correlation windows are illustrated in Figure 2-6. The interrogation window size is 32×32 *pixel*. Gradient threshold for compression is set at 20 grayscales. Notice that a number of edges are positioned across neighboring windows in the second image. The measured sparse disparity field has a mean of 7.67 *pixel* and a standard deviation of 1.56 *pixel*.

The measured disparity results based on adaptively selected correlation windows are illustrated in Figure 2-7. Interrogation window size is still 32×32 *pixel*. Gradient threshold for compression is also 20 grayscales. Each cross-correlation window location is determined based on the edge map of the second image. Note the significantly improved accuracy with adaptive windows. The measured disparity field of valid vectors has a mean of 7.99 *pixel* and a standard deviation of 0.0505 *pixel*. The number of valid vectors is 50, and the number

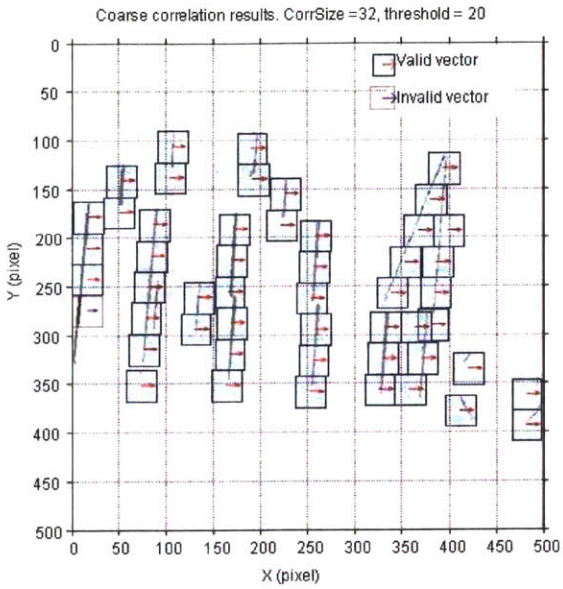

Figure 2-7: Compressed coarse correlation results of an image pair with a simulated horizontal disparity of 8 *pixel* using adaptive windows. Correlation window location is determined based on the edge map of the second image.

of invalid vectors is 1. The time required for both block finding and compressed correlation is 7.5 ms on a Xeon 2.8GHz desktop using a C++ implementation.

For the purpose of statistical analysis, a stringent vector validation threshold is set at ± 0.5 pixel from the true disparity. Any measured disparity that falls outside this tolerance range is classified as an outlier; otherwise, a valid vector. For example, if the true disparity = 8 *pixel*, the range of valid measured vectors is 7.5-8.5 *pixel*. If the true disparity = 1 *pixel*, valid range of measured vectors is 0.5-1.5 *pixel*. The only invalid vector in Figure 2-7 is due to the image boundary effect. Gross errors occur when some edges are entering or leaving the field of view between exposures. This issue is probably unsolvable with only two images. Figure 2-8 shows both the original images and extracted edge maps of the one invalid correlation window in Figure 2-7. The measured disparity is 5.91 *pixel* compared to the true disparity of 8 *pixel*.

There are two ways to automatically detect and discard such spurious correlation vectors. The first method only applies to correlation windows sitting across image boundaries by excluding a pre-defined large boundary area from edge detection. The second method applies

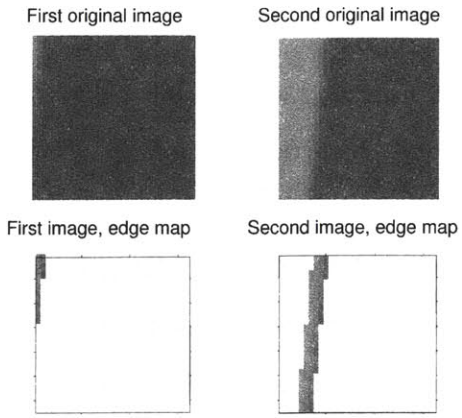

Figure 2-8: Demonstration of the image boundary effect. The top shows the original images and the bottom their corresponding extracted edge maps.

to correlation windows over the entire reference image including boundary area, which is explained in the following section.

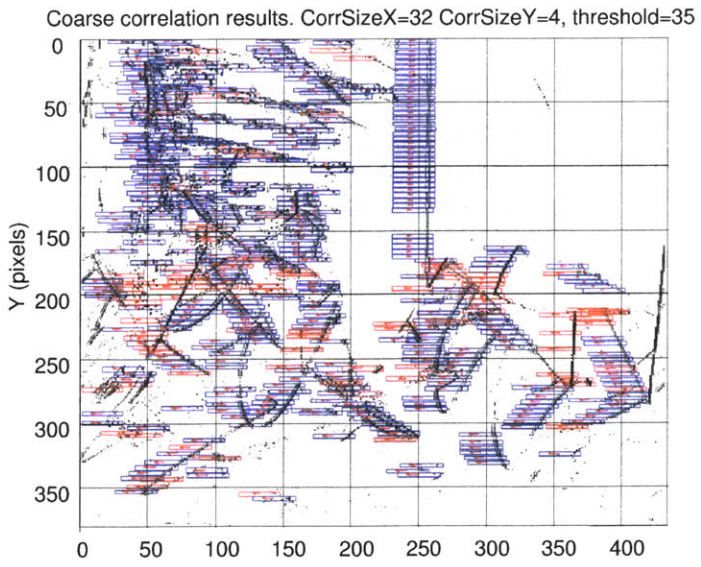

Figure 2-9: Erroneous coarse edge correlation results for the "Sawtooth" image pair.

2.6 Confidence measure

Adaptive window alone still cannot guarantee reliable cross-correlation vectors. Figure 2-9 shows the correlation outputs of the "Sawtooth" image pair when adaptive window is used. Blue windows represent valid correlation vectors, while red windows invalid ones. Validity is determined by whether the vector output is within ± 1 pixel range of this reference window's groundtruth, which is calculated by averaging compressed pixels' true disparities. A less stringent validity threshold is used here to conform to conventions.

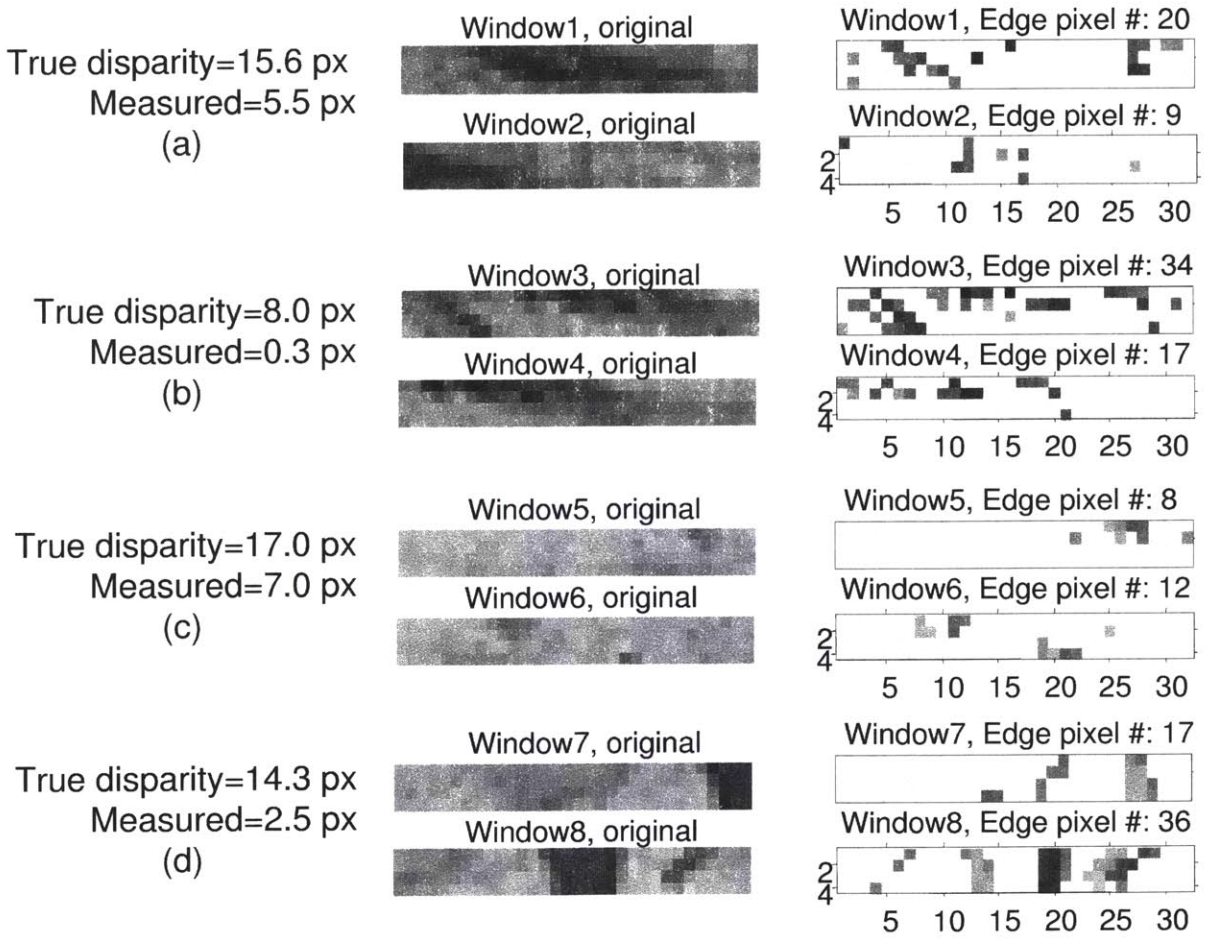

Figure 2-10: Original and compressed images of four sets of erroneous windows.

After studying the erroneous windows, it is noticed that gross errors often occur in areas with dense features. Figure 2-10 illustrates four of such window pairs. In each set

of examples, the left column compares the groundtruth and measured disparity values of this window pair. Middle column is the original images. Window #1 shows the correlation window in the first reference image chosen by the adaptive window measure. Window #2 shows the contents of the same correlation window in the second image where all pixels shift to the left. Right column shows the distribution and total number of pixels left after compression.

We can notice that in Figure 2-10 (a) and (b), the left part of the edges went out of the correlation window in the second image. As a result, the total number of compressed pixels is smaller in the second window. In Figure 2-10 (c) and (d), more features enter the second correlation window from the right. Consequently, the total number of compressed pixels is larger in the second window. In both scenarios different number of edges are present in the correlation window pair, which leaves a large discrepancy in the total number of compressed pixels.

This problem cannot be solved by extending search ranges at the right hand of the first image or the left hand of the second image, particularly not helpful in densely featured areas. Increasing window size may lessen the problem to some extent. However more errors will be generated due to severe averaging effects in depth discontinuity regions.

Thus a confidence measure is introduced to select reliable correlation windows. About the same number of edges should appear in both cross-correlation windows to produce a reliable output vector. This requirement is easy to check in compressed format because the number of compressed pixels is directly proportional to the amount of features. The following equation defines the confidence measure formula:

$$|numCor1 - numCor2| < min(\mu, min(numCor1, numCor2) \times \lambda)) \tag{2.8}$$

where numCor1 and numCor2 represent the total number of compressed pixels in the two correlation windows.

Only when the absolute difference between the numbers of compressed pixels is smaller than a threshold, should this window pair's output be kept and marked as valid. The

threshold is chosen between a fixed number μ and a fraction of the smaller compressed pixel count, whichever is smaller. The parameter μ is selected based on the window size. For example, μ=10 works well for a 32×4 *pixel* window size. λ is usually set to 0.5.

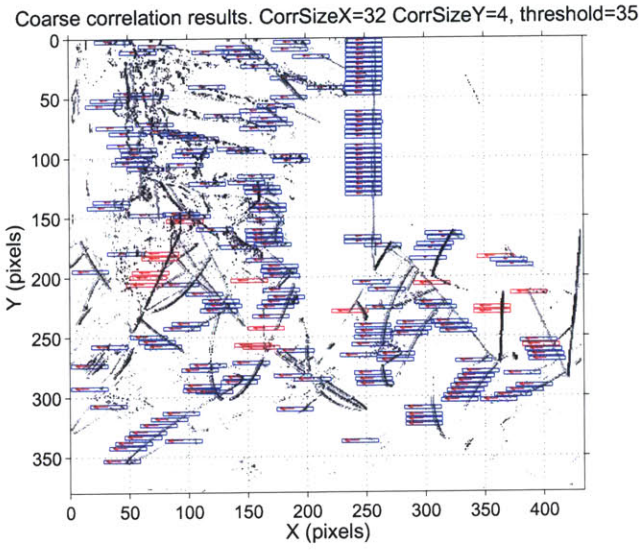

Figure 2-11: Coarse edge correlation results for the "Sawtooth" image pair using the confidence measure.

The four erroneous windows in Figure 2-10 are now discarded under the confidence measure. The differences of compressed pixels in (a), (b), and (d) are larger than 10 pixels. The difference in (c) is equal to half of the number of compressed pixels in the first image.

Correlation results with confidence measure on the "Sawtooth" benchmark image are shown in Figure 2-11. Figure 2-12 compares the error rate histograms before and after applying the confidence measure. All compressed pixels are assigned the same disparity value of the interrogation window that they belong to. Validity of one compressed pixel's disparity is determined by whether its assigned disparity is within ± 1 pixel range of its groundtruth. Total error rate is calculated by dividing the number of invalid pixels over the total number of compressed pixels, while discontinuity error rate only considers the depth discontinuity areas. Number of erroneous pixels are shown in logarithmic scale. Invalid pixels are counted in different buckets based on their deviation from groundtruth. It is obvious that correlation error rate is significantly reduced by around 10% in both smooth

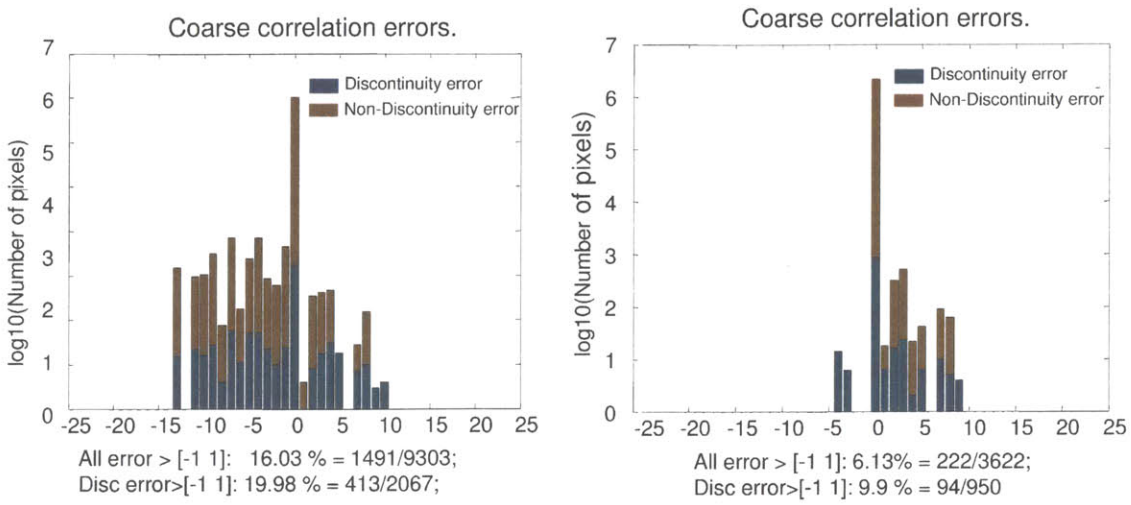

Figure 2-12: Coarse edge correlation error rate with and without the confidence measure.

and discontinuous regions with the help of the confidence measure.

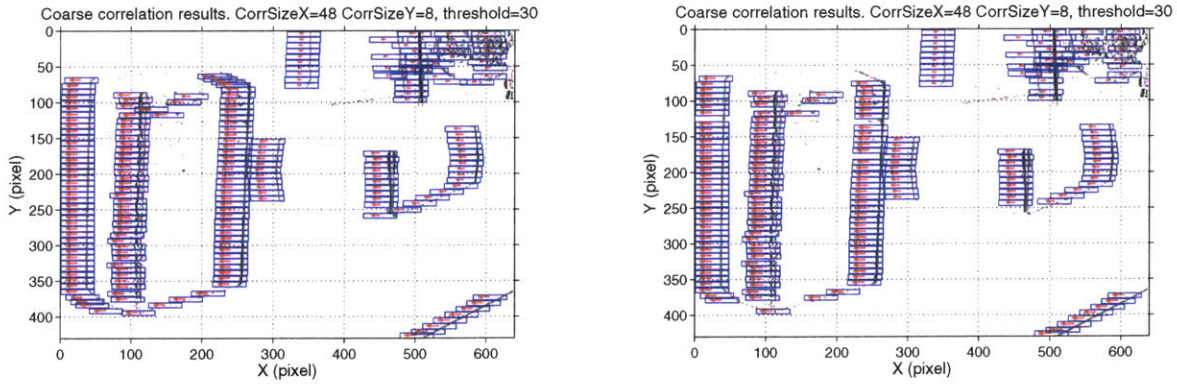

Figure 2-13: Coarse correlation results from the "Box" scene. Left: without confidence measure; Right: with confidence measure.

The above confidence measure is a very stringent one. From Figure 2-12, we can see the number of compressed pixels in the remaining windows has dropped by more than one half. If we compare Figure 2-9 and Figure 2-11, it is obvious that many valid blue windows in densely featured areas are also discarded because they fail to pass the confidence measure. Still in order to improve robustness it is more important to weed out gross errors than keep multiple valid windows in a small area, especially when the output of Compressed Feature Correlation will be used as prior information to graph cut.

There are still some invalid red windows left in Figure 2-11. All these errors are caused by the averaging effects of local windowed approaches. When a correlation window sits across the foreground and background objects with a large disparity step, the measured disparity vector is somewhere in the middle. This problem is inherent of local matching methods.

Confidence measure hardly affects the results of sparsely featured images, as illustrated by Figure 2-13. In this case, μ is set to 20 *pixel* since a window size of 48×8 *pixel* is much larger than the "Sawtooth" example.

In summary, adaptive window and confidence measure are both ways to ensure that the same set of features are present in the two corresponding correlation windows.

2.7 Fine correlation and dense depth map

Coarse compressed correlation provides an averaged disparity estimate for each window. A large window size is necessary in order to accommodate edges of different shapes and orientations in both images. As explained in Section 2.4, cross-correlation generates spurious vectors if an edge is fully present in one correlation window but half missing in the same correlation window in the other image. Usually, the coarse correlation window size is chosen to be roughly twice the size of the largest expected output disparity. In general, a larger window produces fewer errors but it slows processing.

Once a disparity estimate from coarse correlation is known, fine correlation can be performed for each on-edge pixel in the primary correlation window using a much smaller window size. The location of fine correlation window in the second image is set around the pixel of interest. The corresponding fine correlation window in the first image is shifted by the integer amount of the coarse correlation output, as illustrated in Figure 2-14. Here, fine correlation window size is chosen to be 7×7 *pixel*. As a general rule, fine correlation speed dramatically improves with reduced fine correlation window size, at a price of reduced accuracy.

Other aspects of the fine compressed correlation are the same as coarse correlation except a lower gradient threshold. The compression threshold is lowered to avoid the loss of any useful information in the reduced correlation window. This practice does not compromise

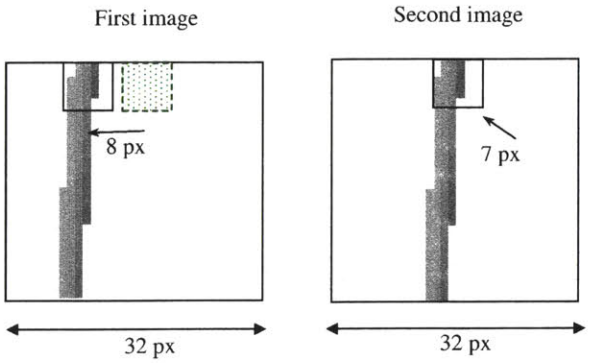

Figure 2-14: Example of a fine correlation window selection in a coarse correlation window with a calculated disparity of 8 *pixel*. The pixel of interest is at the center of the fine correlation window in the second image. The corresponding fine correlation window in the first image is shifted to the left by 8 *pixel*. Fine correlation window size is 7×7 *pixel*.

robustness since a valid disparity has been identified using a higher $C\_threshold$ in coarse correlation. Figure 2-15 illustrates the fine correlation results up to integer-pixel resolution based on the coarse correlation output shown in Figure 2-7. Fine correlation has an improved accuracy over coarse correlation because of window shifting. In this simulated disparity case, the disparity of every on-edge pixel is correctly recovered with a standard deviation of 0 *pixel*.

In some applications such as 3D feature-based facial recognition [33] and object tracking [27], a sparse disparity map of object boundaries provides enough information. In other applications such as 3D scene reconstruction, a complete disparity map is preferable. There are two general approaches to fill the regions of unknown disparities among the recovered edges. If there is sufficient surface texture, compressed cross-correlation at a lower threshold can be performed in such areas. However, in many real world cases such as the scene in Figure 2-1, there is a lack of fine texture, which is critical to a reliable correlation. Instead, depth-driven object segmentation and depth interpolation is necessary to obtain a full field disparity rendering.

The overall flow of the proposed edge-matching algorithm with complete disparity map output is shown schematically in Figure 2-16.

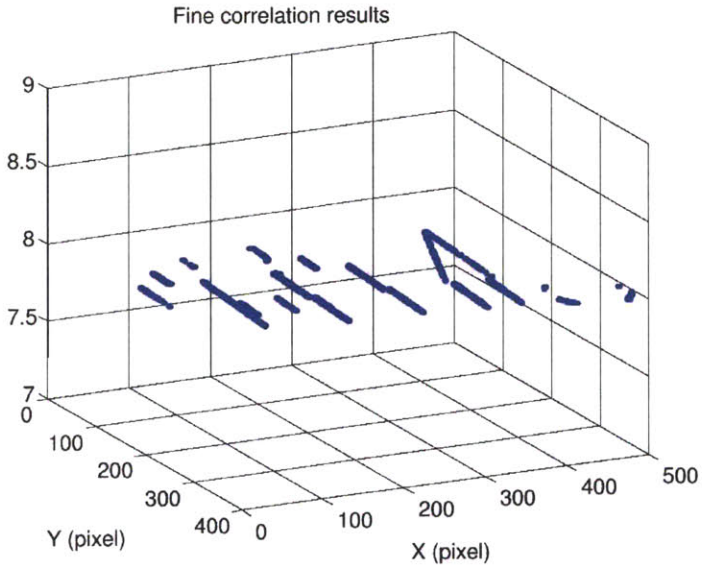

Figure 2-15: Sparse fine correlation results up to single pixel resolution of an image pair with a simulated disparity of 8 *pixel*. $C\_threshold = 15$ *grayscales*. The sparse disparity field has a mean of 8 *pixel* with a standard deviation of 0 *pixel*.

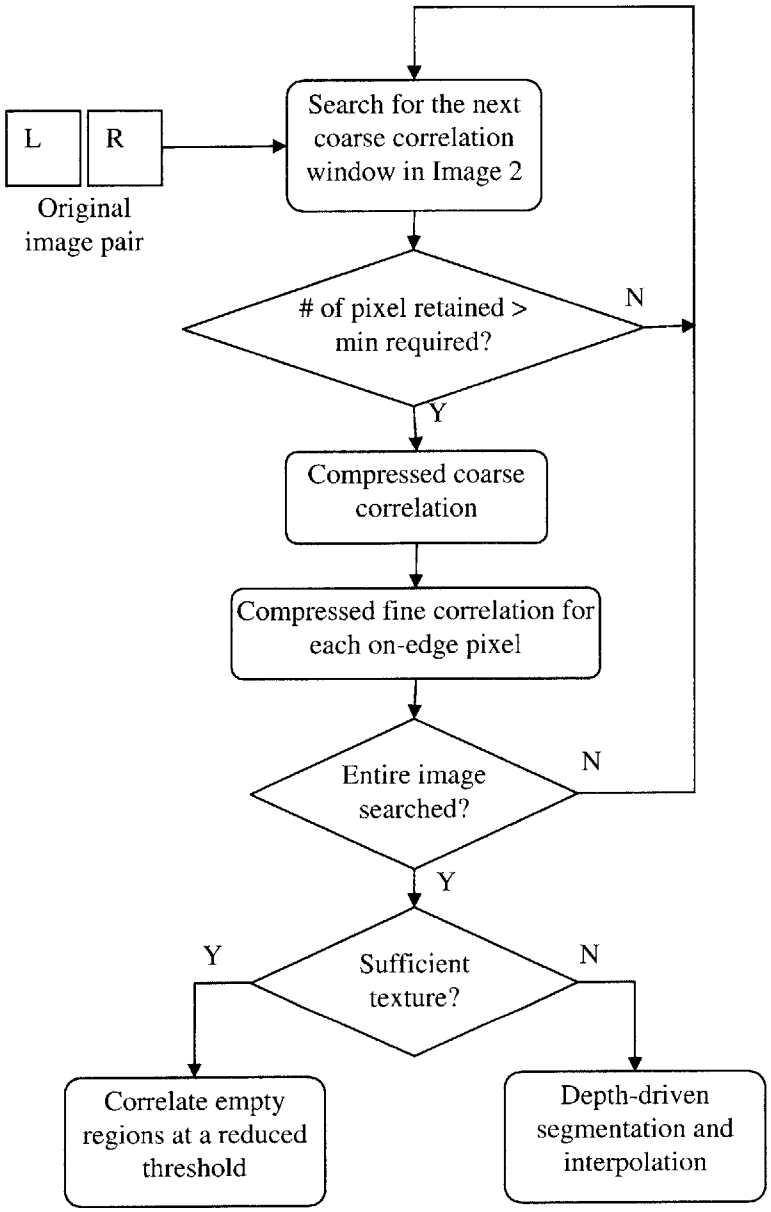

Figure 2-16: Overall flow of the proposed algorithm.

Chapter 3

Performance evaluation of Compressed Feature Correlation

This chapter provides qualitative and quantitative experimental results on both simulated disparities and real image pairs. The processing speed and quality of the disparity maps obtained demonstrate the effectiveness of the proposed Compressed Feature Correlation algorithm in Chapter 2. The algorithm's sensitivity to parameters is discussed in Section 3.3.2.

3.1 Simulated disparity

The original image in Figure 2-5 is artificially shifted laterally to the right with a disparity value from 0.2 to 11 *pixel* at an interval of 0.2 *pixel*. Sub-pixel disparity is approximated using the shift theorem in frequency domain [78]. Fine correlation is calculated at each on-edge pixel. Figure 3-1 shows the standard deviation of fine correlation results throughout the entire sequence of lateral disparities. The maximum standard deviation among integer disparities is ± 0.0812 *pixel*. The maximum error among simulated sub-pixel disparities is ± 0.147 *pixel*. Overall mean value of standard deviations is ± 0.098 *pixel*. Given the pixel intensity rounding error introduced in the sub-pixel shifting simulation, it is fair to conclude that the proposed algorithm has an accuracy upper limit of ± 0.1 *pixel*, which is consistent

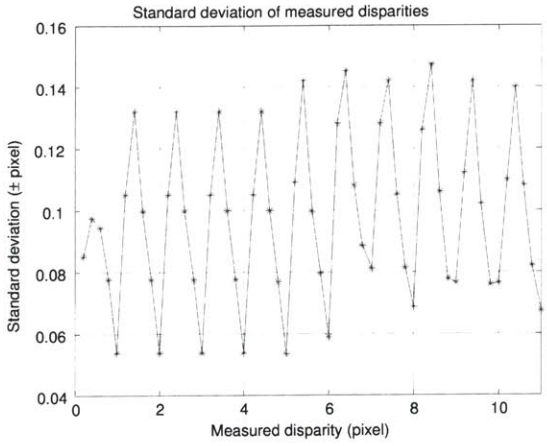

Figure 3-1: Standard deviation of measured disparities of a sequence of images with a simulated horizontal disparity from 0.2 to 11 *pixel* relative to the original image. Coarse correlation window size = 32×32 *pixel*. Fine correlation window size = 7×7 *pixel* and $C\_threshold = 15$ *grayscales*.

with the best obtainable accuracy from interpolating a single correlation plane calculated with only two images. The periodic structure in Figure 2-5 is related to pixel discretization error during shifting simulation.

3.2 Qualitative results on real images

Real image pairs are better test-beds for stereo vision algorithms because systematic and random errors are an inevitable reality from frame to frame. Possible error sources include optical distortion, reflectivity, lighting fluctuation, occlusion and camera dark noise.

The results on the following four image pairs demonstrate the algorithm's effectiveness at accurately capturing depth discontinuities in addition to other dense intensity features. Compressed Feature Correlation correctly finds disparity values for the dominant features in images.

3.2.1 Coarse correlation

The algorithm's outputs on four different stereo image pairs are presented. The first two are taken in a typical room with a single Sony digital camera which was translated along a baseline of 10 *mm*.

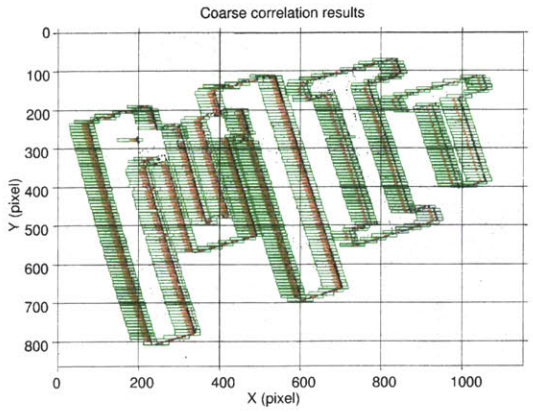

Figure 3-2: Coarse correlation results of the "MIT" image pair with lateral disparities. CorrSizeX = 64 CorrSizeY = 8, $C\_threshold = 15$.

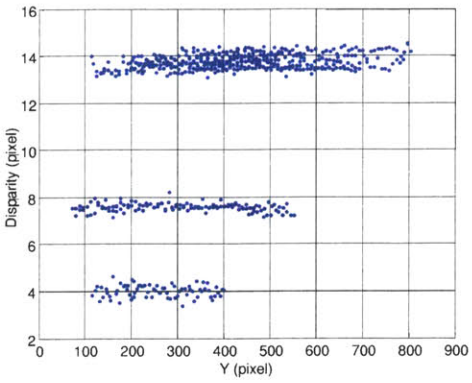

Figure 3-3: Side view of the coarse correlation disparity map shown in Figure 3-2.

In the following "MIT" scene shown in Figure 3-12, both images in the image pair are captured by a moving camera. The image size is 1152×864 *pixel*. The camera has a lateral leftward displacement between the two exposures resulting in right-hand disparities in the image plane. The three objects are placed on various depth planes. The closer the object is

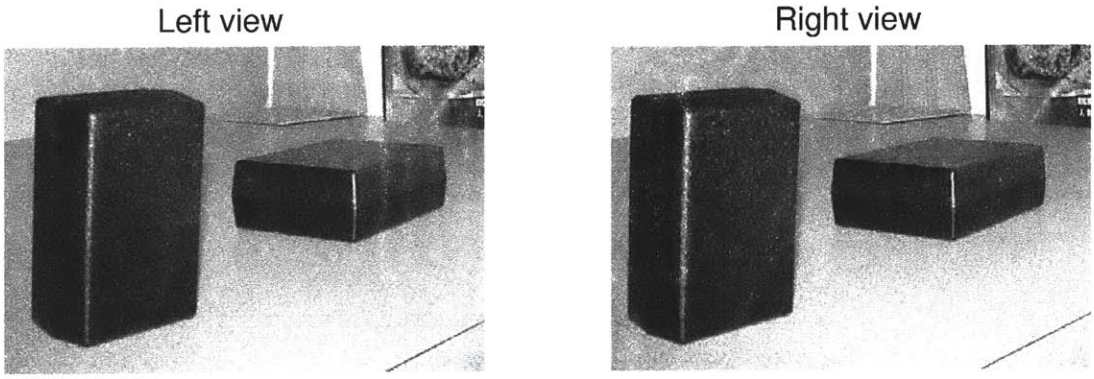

Figure 3-4: "Box" image pair

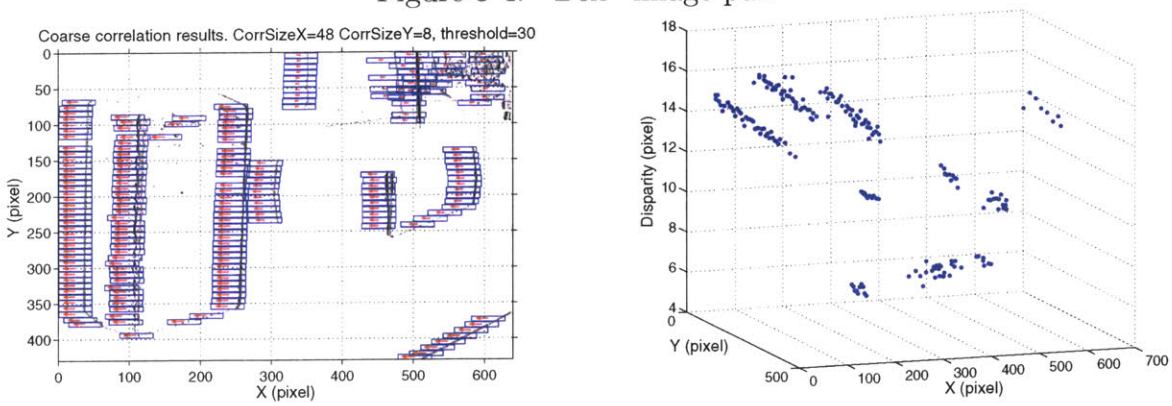

Figure 3-5: 2D and 3D renderings of coarse correlation results from "Box" pair.

to the camera, the larger its disparity between frames. Choosing a proper coarse correlation window size is critical to the proposed algorithm's performance on speed and accuracy. Overall, a larger window size produces fewer gross errors but it slows processing because the correlation load increase with respect to the square of the window size. On the other hand, a smaller correlation window size results in higher spatial resolution and thus less averaging effect in areas where a number of objects at different depths are close to each other. In the "MIT" scene, a rectangular window shape is able to take advantage of both large and small window sizes since disparities are known to occur in only one direction. The horizontal correlation window width is chosen to be 64 px, while the vertical window height 8 *pixel*.

Figure 3-2 shows the vector field of coarse correlation results as well as their adaptively

selected corresponding windows. Figure 3-3 provides a side view of the coarse correlation disparity field. Disparities of all major object boundaries are correctly recovered. The processing time of both block-finding and coarse correlation is 29.0 *ms* on a Xeon 2.8GHz desktop for an image pair with a size of 1152×864 *pixel*.

Figure 3-4 shows the "Box" image pair where the left view is the reference image. Image size is 640×430 *pixel*. Two boxes sit on a slanted table surface. The background is two tilted perpendicular walls. This scene has very sparse features except a small area on the wall. Figure 3-5 gives top and side view of its coarse correlation results. The edges of two boxes are clearly separated from each other as well as the background and foreground table rim.

The following examples are two standard indoor scenes from [9] where all disparities are to the left, and left views are the reference images. The level of features is much denser than the previous two examples and the topology is more complex. Figure 3-6 shows the "Room" pair, where the foreground table and equipment have a disparity around 13 *pixel* while the far background conference a disparity of $1-2$ *pixel*. Figure 3-7 gives top and side view of its coarse correlation results. Major edges' disparities are robustly identified.

Figure 3-8 shows the "Lamp" pair, where the foreground lamp has a disparity around 13 *pixel* while the background boxes a disparity around 7 *pixel*. The slanted optical table surface does not have strong features. Figure 3-9 gives top and side view of its coarse correlation results. Again, strong features' disparities are reliably recovered in both sparsely and densely featured regions. Note that in the reference image, left and right boundaries of half the correlation window width are excluded from edge detection in order to avoid boundary errors.

3.2.2 Fine correlation and dense depth map

For a special type of scenes where each object's boundary is clearly defined and each object can be closely approximated as a frontal-parallel, slanted or curved surface, the sparse depth map could be enough to generate a full field dense disparity map. The "MIT" scene is one such example. Each of the three letters sits on a slanted surface and disparities of their

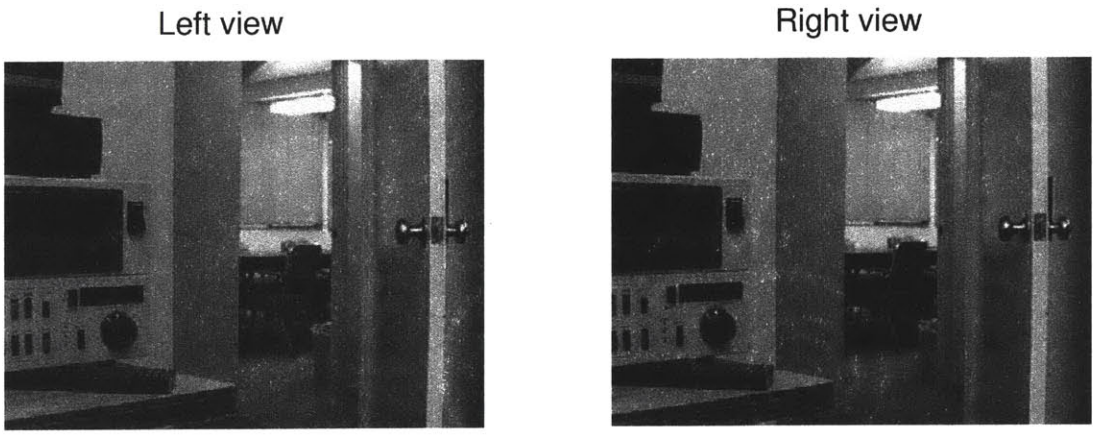

Figure 3-6: "Room" image pair

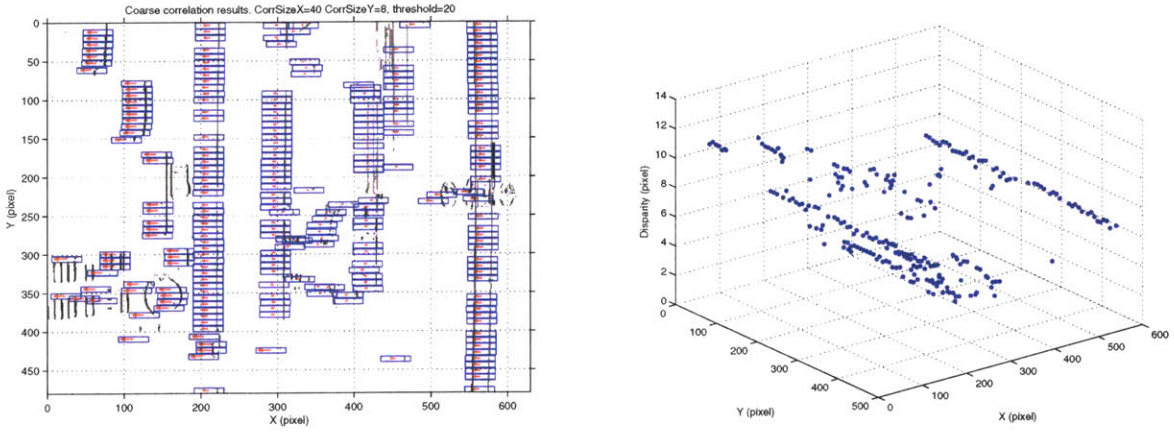

Figure 3-7: 2D and 3D renderings of coarse correlation results from "Room" pair.

mostly enclosed boundaries can be obtained from fine compressed correlation. Figure 3-10 and Figure 3-11 illustrates both the top and front views of the fine compressed correlation disparity field comprised of all the detected on-edge pixels.

After a fine disparity map of the object boundaries is computed, there are two general approaches to generate a dense disparity map. The first method is to fit a surface for each set of boundaries. For this example affine motion fitting is enough. Higher order quadratic fitting will be necessary for curved surfaces. The second method is a simple segmentation and 3D interpolation method, which is faster than the fitting approach since no filtering or

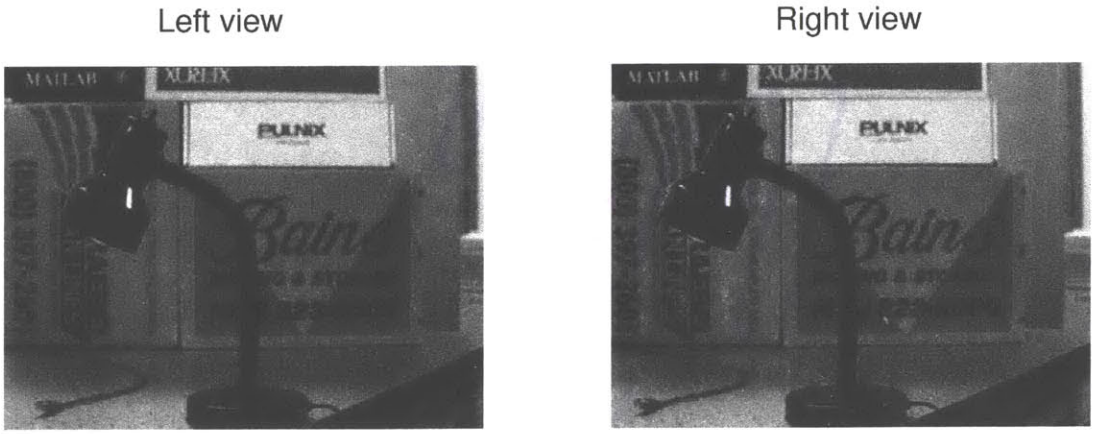

Figure 3-8: "Lamp" image pair.

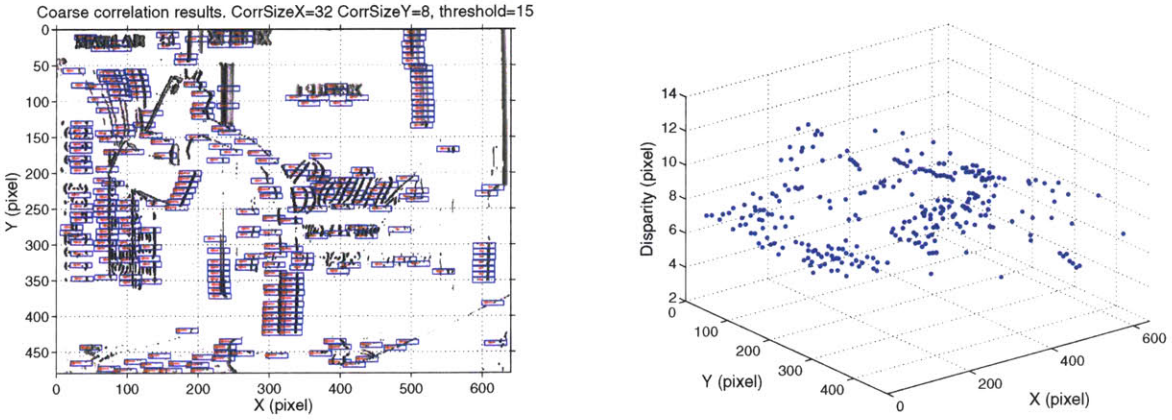

Figure 3-9: 2D and 3D renderings of coarse correlation results from "Lamp" pair.

minimization is needed to further improve the accuracy. In this demonstration, one such algorithm based on the second approach is implemented to fill the voids among the object boundaries because there is no sufficient texture on their smooth surfaces for a reliable correlation. Figure 3-12 shows the full disparity field rendering of the "MIT" scene both with and without texture mapping. The results are encouraging considering only a single image pair is used as input. A 3D depth map rendering may be obtained using Equation 1.3 after calibrating the moving camera setup.

If there is sufficient texture on smooth surfaces, compressed correlation over the entire

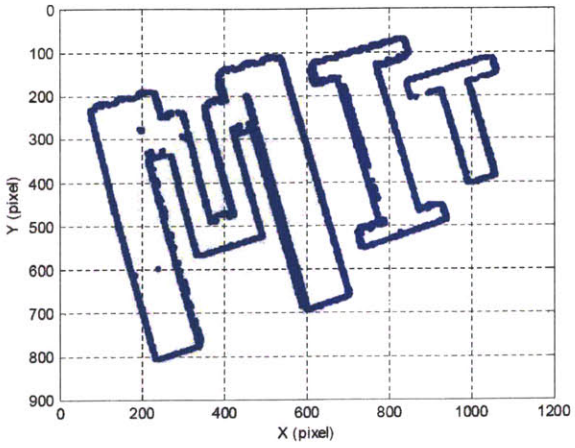

Figure 3-10: Top view of fine correlation results calculated based on the coarse correlation output shown in Figure 3-2.

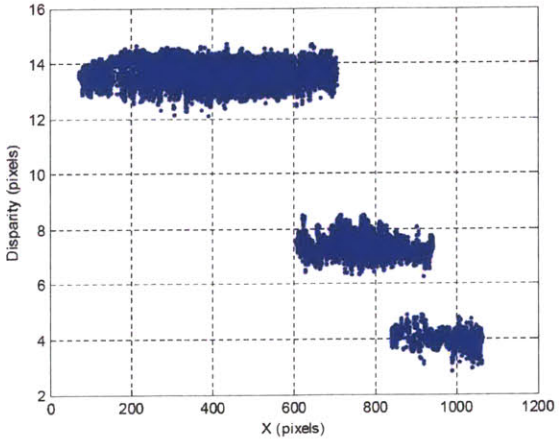

Figure 3-11: Front view of fine correlation results calculated based on the coarse correlation output shown in Figure 3-2.

image plane can be performed to obtain a complete disparity map. Two different levels of compression may be implemented. In boundary regions, a strong compression is applied which results in the precise recovery of edges. In regions of small gradient variations, which often correspond to smooth surfaces, a mild compression is held, in which case any useful information for correlation is retained including minor features in surface texture.

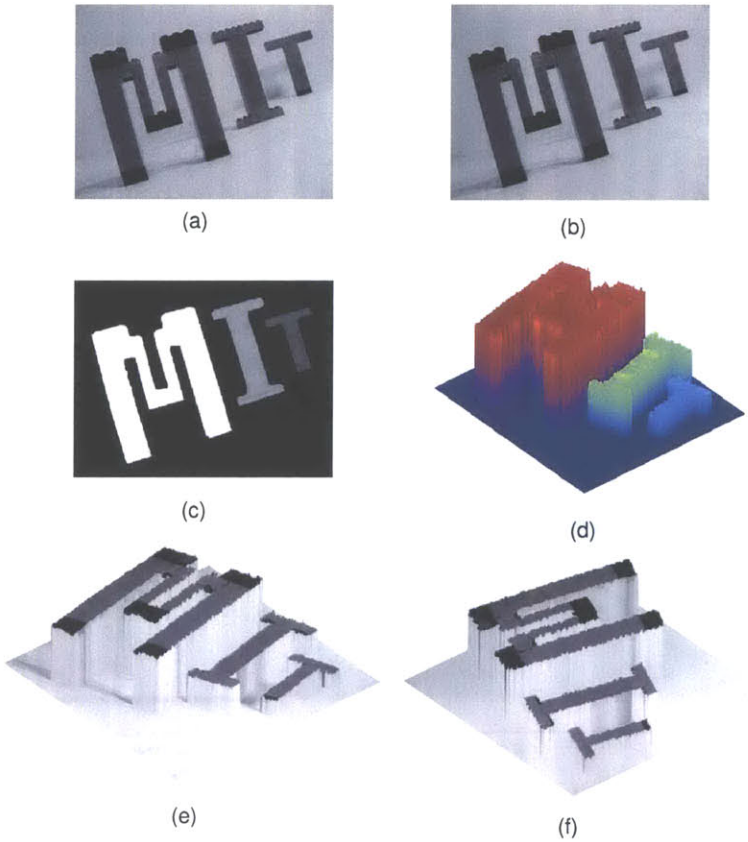

Figure 3-12: Full disparity map rendering of the "MIT" scene. (a) left image; (b) right image; (c, d) top view and 3D rendering of the complete disparity map after segmenting and interpolating the sparse correlation output shown in Figure 3-10 and Figure 3-11; (e, f) two views of the complete disparity map with texture mapping.

3.3 Quantitative results on benchmark images

Next the Compressed Feature Correlation is tested on four well-known Middlebury benchmark images [114]. Figure 3-13 gives the left view reference image of each pair. Three evaluation masks are also given along with groundtruth: non-occluded, discontinuous and untextured regions are represented by white areas. Only the non-occluded masks are listed in Figure 3-13, which are used for calculating total error rates. These four stereo pairs encompass a variety of situations that are most representative in computer vision, such as textured or untextured objects or backgrounds, planar or curved surfaces, frontal-parallel or

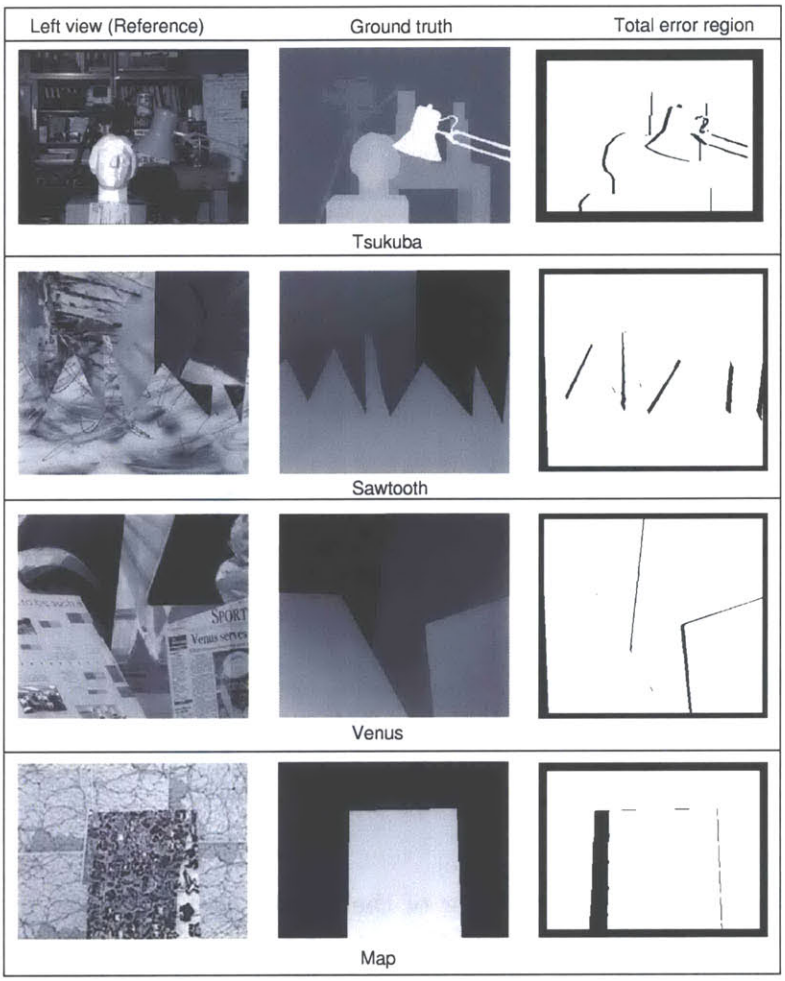

Figure 3-13: Four sets of benchmark images.

slanted surfaces, small or dramatic depth discontinuities. Note that compared with "MIT" and "Box" scenes, these images are much more densely textured.

Figure 3-14 to Figure 3-17 demonstrates the coarse correlation results using Compressed Feature Correlation. Each background image shows the compressed reference image with $C\_threshold = 35$ *grayscales*. Blue and red windows highlight the location of valid and invalid correlation windows, respectively. Correlation window height is fixed at 4 *pixel*, while window width is proportional to disparity search length Δ. Confidence measure parameter μ is set to 10 for all four image pairs for simplicity.

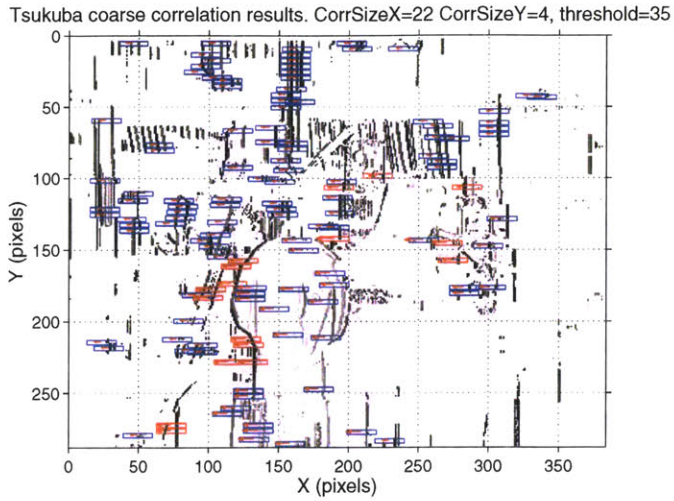

Figure 3-14: Coarse correlation results from "Tsukuba" pair.

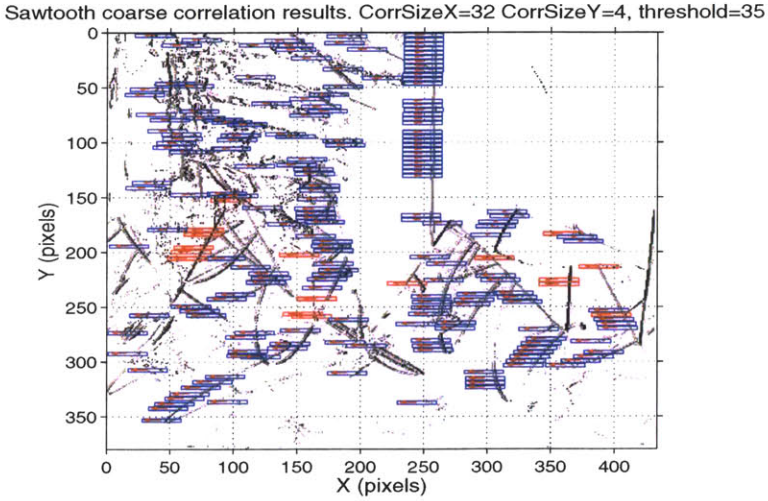

Figure 3-15: Coarse correlation results from "Sawtooth" pair.

Some correlation windows around feature points are invalidated by the confidence measure, such as the scenerios explained in Figure 2-10. Also, even when the same intensity feature is present in both corresponding windows, this correlation window might be eliminated if the contrast of the feature changes significantly from frame to frame and consequently the window cannot pass the confidence measure. As a result, only strong and constant features contribute to the correlation outputs. If we take a look at the red invalid windows, they all occur around depth discontinuities or occluded areas as explained in Chapter 2.

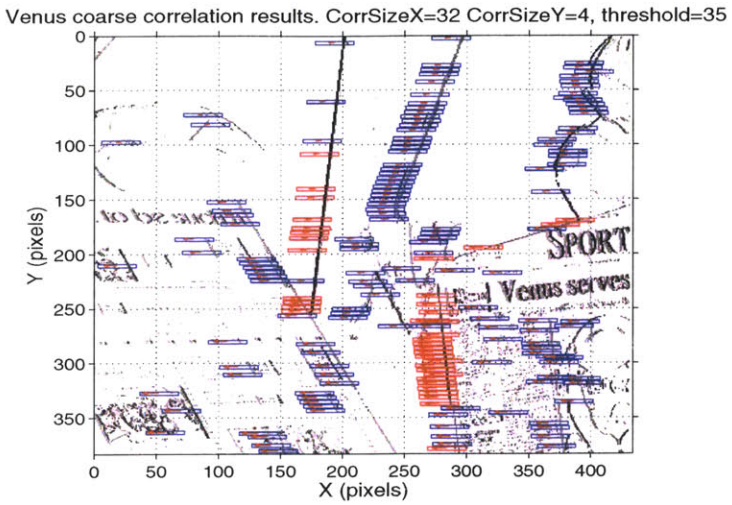

Figure 3-16: Coarse correlation results from "Venus" pair.

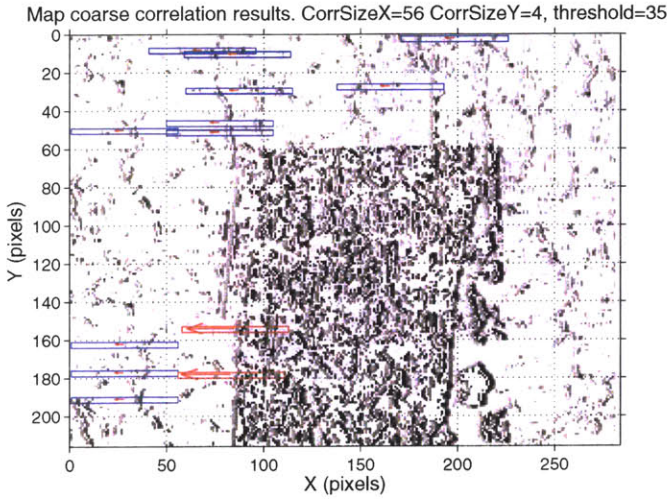

Figure 3-17: Coarse correlation results from "Map" pair.

3.3.1 Computing time

Compressed Feature Correlation is extremely fast in terms of estimating the disparities of strong features. The processing time in Table 3.1 is tested on a laptop equipped with a 2 GHz Intel Pentium 4 CPU and using VC++ environment. As a general rule, processing time is proportional to image size, feature density, cross-correlation window size, correlation search length, and compression threshold. The "Map" pair takes the longest time despite of its smaller image size because its correlation window size is 75% larger than "Sawtooth"

Stereo pair	Image size (*pixel*)	Search range 0-Δ (*pixel*)	Window size (*pixel*)	Time (*ms*)
Tsukuba	384×288	0-15	22×4	5.8
Sawtooth	434×380	0-18	32×4	9.5
Venus	434×383	0-20	32×4	9.3
Map	284×216	0-28	56×4	21

Table 3.1: Computational time on four benchmark images.

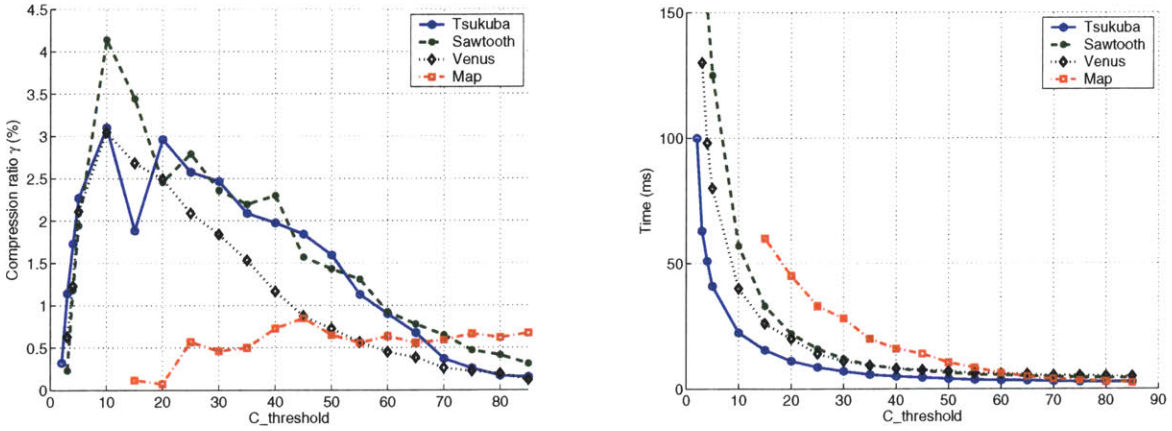

Figure 3-18: Left: Compression threshold *vs.* compression ratio; Right: Compression threshold *vs.* processing time.

due to its long search range Δ. 95% of detected windows are discarded under the confidence measure. These discarded windows do not go through the entire correlation process, but they still take up compression time.

The compression threshold *C_threshold* significantly influences processing time as shown in Figure 3-18. Very few or even no valid windows are detected when $C\_threshold \leq 5$ because the compressed features are too dense. Low spatial frequency components are most dominant in real images. High local gradient areas become sparser as *C_threshold* goes up and consequently less pixels to compress and correlate. The number of compressed pixels decreases when $C\_threshold > 10$ since minor features are ignored and less pixels from a strong feature are kept. Compression ratio γ is defined as the number of remaining pixels after compression divided by the total number of pixels in the image. γ starts to drop below 0.5% when $C\_threshold \geq 60$. Speed improves by $75-90\%$ when *C_threshold* increases from 15 to 60 grayscale.

3.3.2 Sensitivity to parameters

Description	Symbol	Value
Compression threshold	*C_threshold*	35 *grayscale*
Confidence measure fraction	λ	0.5
Confidence measure threshold	μ	10 *px*
Correlation window width	w	22, 32 or 56 *pixel*
Correlation window height	h	4 *pixel*

Table 3.2: Parameter settings of compressed feature correlation.

For all the results presented in Section 3.3, the parameters are set to the same values in Table 3.2 if not mentioned otherwise. Correlation window width w is chosen according to the size of search length Δ. w is set to 22 *pixel* for "Tsukuba", 32 for "Sawtooth" and "Venus", and 56 for "Map". In order to evaluate the algorithm's sensitivity to parameter settings, each parameter is varied in turn while other ones are fixed to their values in Table 3.2. Figure 3-19 demonstrates how total error rate varies with different compression threshold in all four benchmark images. Figure 3-20 presents the relationship between normalized μ and compression ratio as well as total error rates for the "Tsukuba", "Sawtooth", and "Venus" image pairs. Figure 3-23 shows the results of normalized window width *vs.* compression ratio and total error rate for the above three image pairs. Finally Figure 3-24 illustrates how the compression ratio and total error rates are affected by the correlation window height.

The algorithm's accuracy is not very sensitive to the choice of *C_threshold* in a large grayscale range as illustrated in Figure 3-19. Standard deviation of error rates in the range of $C\_threshold \subset [15, 85]$ is between $1.1 - 3\%$, except for "Map" which is 8.9%. *C_threshold* should be chosen based on application requirements such as speed and compression ratio. An empirical selection range is $C\_threshold \subset [15, 85]$ grayscales.

A depth discontinuity below 10 *pixel* such as the case in the three benchmark images other than "Map" is usually suitable for correlation-based methods. "Map" is especially difficult for correlation-based methods because it exhibits a combined characteristic of both dense features and significantly large disparity discontinuities. The negative impact is twofold. First, very few valid correlation windows could pass the confidence measure because different

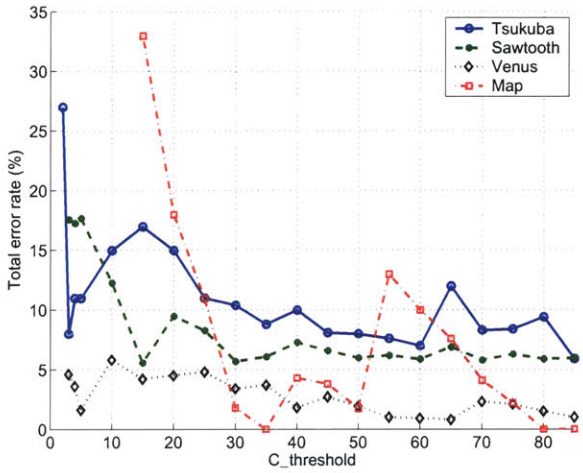

Figure 3-19: Compression threshold *vs.* total error rate.

features come in and leave the big correlation window pair. As a result, the compression ratio for "Map" is well below 1% for various $C\_threshold$ values as shown in Figure 3-18. Second, the few valid windows that sit across depth discontinuities produce gross errors that are averages of the close foreground and far background. Consequently, the correct disparity range is not present in the erroneous correlation outputs, which presents a problem to graph cut as will be discussed in Chapter 5. In conclusion, Compressed Feature Correlation could not generate statistically meaningful results on heavily textured images such as the "Map" pair. Thus we exclude it from the remaining analysis in this section.

Figure 3-20 exhibits the influence of the confidence measure threshold μ, where μ is normalized relative to correlation window width w because w is directly related to the number of features that may be present in one correlation window. The compression ratio rises as μ increases and the confidence measure becomes less stringent. It reaches convergence when μ loses significance and the other term in the confidence measure $min(numCor1, numCor2) \times \lambda$) takes over. The inflection point Ω is different for each image composition, for example, around $\Omega = \mu/w$=0.6 for "Sawtooth", 0.8 for "Tsukuba" and 1.2 for "Venus", where Ω is loosely defined. The relationship between error rates and normalized μ is again dependent on image composition. When normalized μ is smaller than 0.2, the error rate might be especially volatile and the compression ratio is relatively small. The standard deviations of

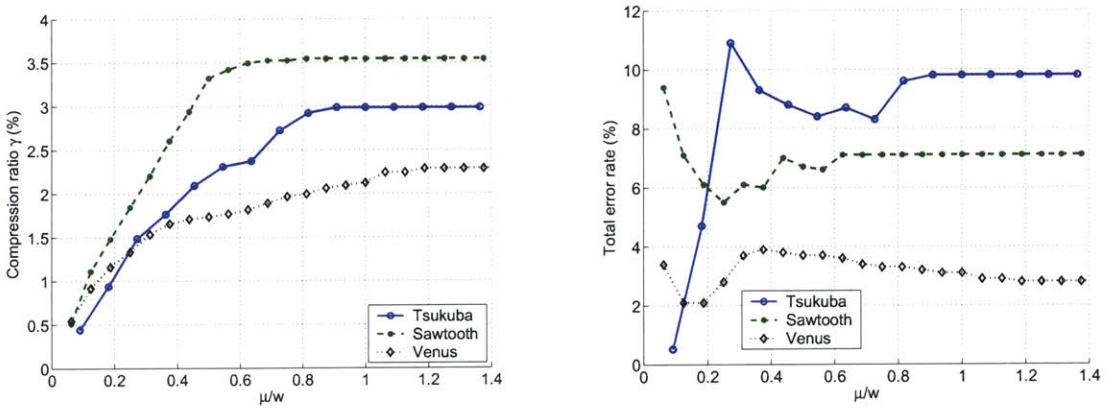

Figure 3-20: Left: Normalized μ *vs.* compression ratio; Right: Normalized μ *vs.* total error rate.

error rate in different normalized μ ranges are listed in Table 3.3, where Ω is set to the values mentioned above.

Normalized μ range	[0.2 Ω]	[0.2 1.4]
Tsukuba	0.90%	0.72%
Sawtooth	0.55%	0.49%
Venus	0.47%	0.46%

Table 3.3: Standard deviation of error rate in different normalized μ ranges.

The fraction value $\lambda \subset [0, 1]$ in the confidence measure Equation 2.8 places a weight on the variable term over the fixed term μ. The larger the value of λ, the less the significance of the variable term. Figure 3-21 illustrates the correlation statistics of "Tsukuba" in three scenarios, where λ is equal to 0.2, 0.5 or 0.8. As λ increases, the variable term becomes less stringent, the inflection point Ω shifts to the right and the converging value of compression ratio increases. If we study the convergence region, higher compression ratio usually leads to more errors in discontinuous areas. However, a small λ is also not preferable because too few valid windows are left to represent a full disparity range. This is why λ is chosen to be 0.5 in this thesis.

Let us consider a simplified version of the confidence measure, where there is no fixed

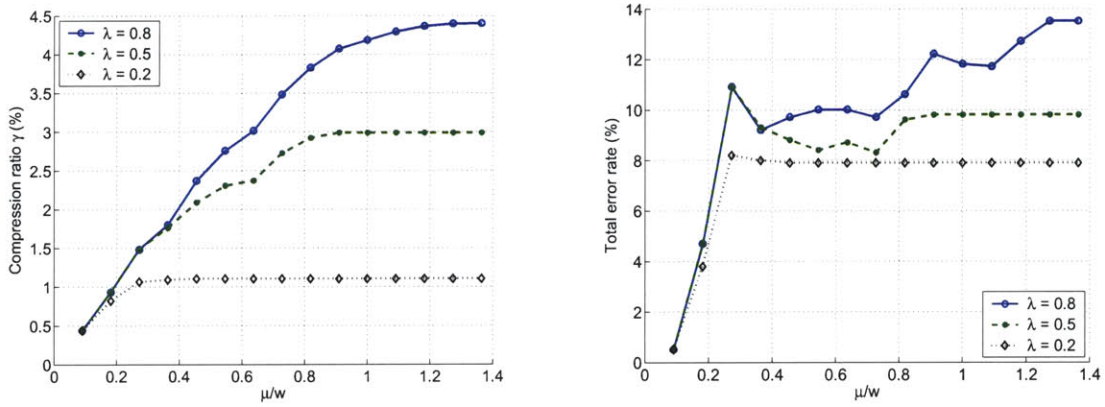

Figure 3-21: Influence of λ on the correlation results of "Tsukuba". Left: Normalized μ *vs.* compression ratio; Right: Normalized μ *vs.* total error rate.

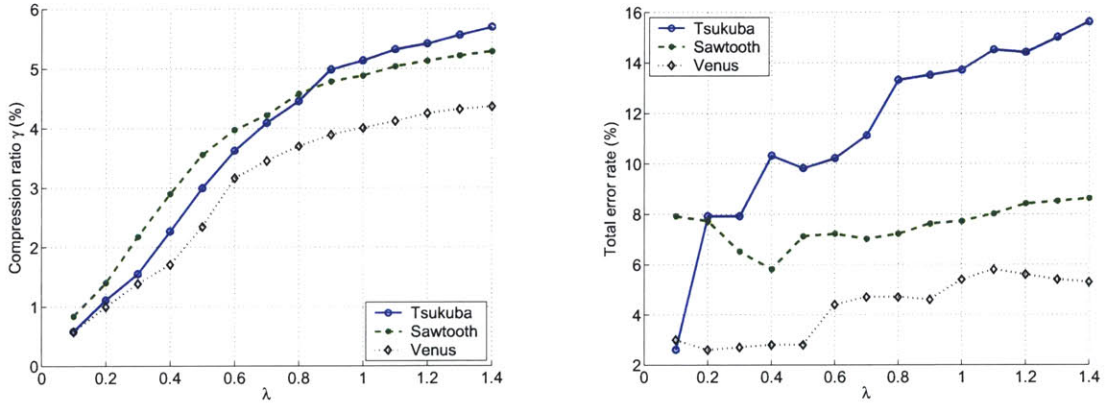

Figure 3-22: Influence of λ on the correlation results of "Tsukuba". Left: λ *vs.* compression ratio; Right: λ *vs.* total error rate.

term μ, or $\mu > w \times h$. Equation 2.8 is then reduced to

$$|numCor1 - numCor2| < min(numCor1, numCor2) \times \lambda \tag{3.1}$$

Figure 3-22 demonstrates the effects of the above simplified confidence measure. Compared to Figure 3-20, the compression ratio rises as λ increases without noticeable convergence in a non-trivial λ range. Consequently, the error rates tend to climb without an upper limit, especially for the "Tsukuba" pair where lots of depth discontinuities and occlusions are present. A robust selection range for λ is $0.4 - 0.6$.

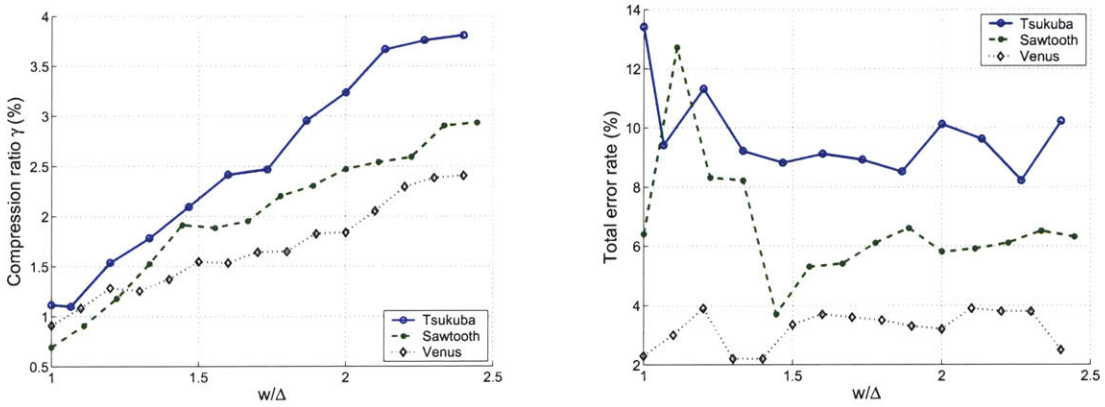

Figure 3-23: Left: Normalized window width *vs.* compression ratio; Right: Normalized window height *vs.* total error rate.

By having two terms, fixed and variable, the confidence measure is more robust. λ controls the upper limits of the compression ratio and error rates, while μ can be flexibly customized for different image types to achieve best accuracy. Thus confidence measure is chosen to be Equation 2.8 in this dissertation.

Figure 3-23 shows how correlation window width w affects accuracy, where w is normalized relative to the maximum search length Δ. As a general rule, w should be at least larger than or equal to Δ in order to detect the largest possible disparity. A large window size usually produces more robust correlation results in the presence of random noises. However a large window size also means more averaging effects around disparity discontinuities and larger computational cost. As w increases, more pixels per window are kept after compression, and thus longer correlation time. Total error rates when the normalized w varies from 1 to 2.5 are presented in Figure 3-23. The standard deviations of error rate in different normalized w ranges are listed in Table 3.4. Empirically, the accuracy is worse when normalized $w \subset [1, 1.5)$ because large disparities might not be fully present in the small window pairs. A range of normalized $w \subset [1.5, 2]$ is often chosen.

Figure 3-24 shows the influence of the correlation window height on accuracy. h is usually chosen to be small in order to improve spatial resolution when there is no disparity in this dimension. $h = 1$ *pixel* gives trivial correlation results because few windows satisfy the

Normalized w range	[1 2.5)	[1.5 2]
Tsukuba	1.43%	0.61%
Sawtooth	2.07%	0.99%
Venus	0.63%	0.19%

Table 3.4: Standard deviation of error rate in different normalized w ranges

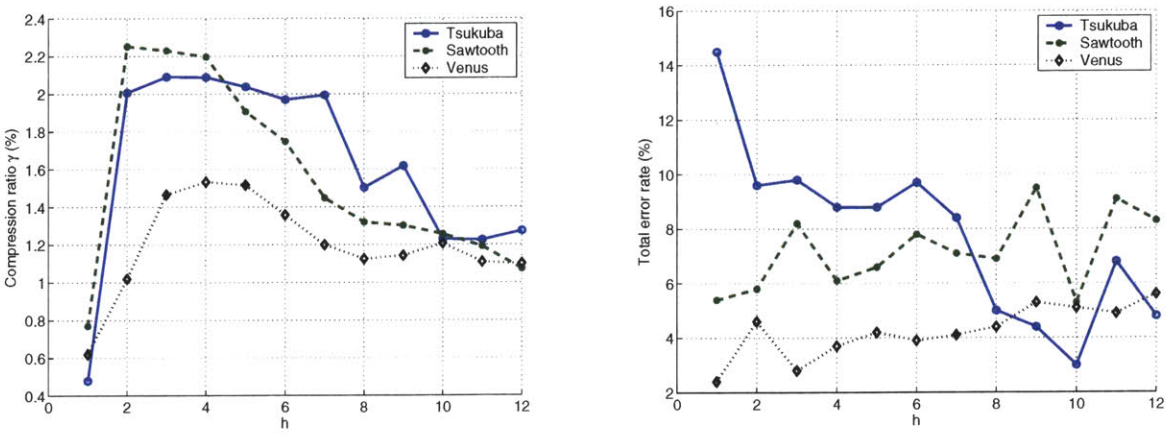

Figure 3-24: Left: Correlation window height *vs.* compression ratio; Right: Correlation window height *vs.* total error rate.

confidence measure due to noise. $h \subset [2, 6]$ renders stable compression ratio and error rates due to the averaging effect in the y-dimension. When $h > 6$ *pixel*, compression ratio drops mainly because less valid windows are detected around disparity discontinuities. At the same time error rates become volatile, whose severity depends on image composition.

In conclusion, the best parameter settings should be chosen considering the combination of image type, computational cost, compression ratio and error rate.

3.4 Summary

In the previous chapter, a new 3D algorithm, Compressed Feature Correlation, has been proposed which can recover precise object boundaries at high speed by utilizing compressed image correlation and adaptive windows. Although the algorithm is relatively simple, the experimental results demonstrated in this chapter are encouraging. An important feature to note is that this algorithm does not include any global optimization.

Compressed Feature Correlation is a technique by which stereo or motion image pairs can be accurately processed at high speeds. It is based on the compression of images in which the number of data set entries is reduced to containing only strong features. Very high correlation speeds are obtained by encrypting the reduced data set into sparse arrays and correlating the data entries using an error correlation function to eliminate time consuming multiplication, division and floating point arithmetic.

The speed performance of compressed image correlation, however, is largely dependent on image complexity. For applications requiring extremely high speeds such as real-time tracking and video rate stereo vision, the proposed feature-based 3D algorithm appears to be a viable processing technique.

Future work includes integrating adaptive window shape and size into Compressed Feature Correlation, as well as multiple image pairs. An extensive analysis of error rate *vs.* spatial frequency would fully demonstrate the algorithm's effectiveness and limitations.

Chapter 4

Graph cut with priors

Compressed Feature Correlation discussed in Chapter 2 and 3 is a local approach and generates a sparse disparity map. This chapter introduces a new graph cut energy model that accepts the results of Compressed Feature Correlation as prior information. The final outputs are dense disparity maps.

This chapter has three parts. First the fundamentals of graph cut are reviewed in Section 4.1. Second part gives examples of three major problems with the standard energy model. Section 4.3 proposes a hybrid graph cut approach based on a modified energy model that takes prior knowledge into consideration.

4.1 Introduction

For years, vision researchers have computed 3D correspondence by averaging all the disparities in a small local window, which can be done efficiently using cross-correlation when sufficient amount of surface texture is available either inherently or by pattern projection during image capture. This is the basic approach taken in Chapter 2 and 3 of this thesis. Each pair of correlation windows generates a real-valued disparity estimation. Unfortunately, active projection is impractical when the dimensions of the objects or environment exceed a certain range due to illumination source energy limitations. Correlation-based methods

generally perform poorly in untextured regions and tend to blur across depth discontinuities.

Dynamic programming stereo methods address the above two issues by searching over all the possible disparities along a scanline [98, 122]. Unlike local methods, here disparities are discretized in order to limit the scope of search. Dynamic programming in stereo is a type of intermediate approach between local and global methods. A scanline may be viewed as a "window" with the height of one pixel and width of the image in the case of rectified images. Rather than assigning an averaged disparity for the whole window, each pixel has its own disparity value. "Snake" methods are based on active couture models using modified dynamic programming [41, 76]. Although very efficient, dynamic programming cannot propagate information among scanlines. Extensive post-processing is necessary to clean up noisy outputs [8]. Recent progress in two-pass dynamic programming reduces inter-scanline inconsistencies by performing optimizations both along and across scanlines [67].

Unlike dynamic programming techniques, global optimization methods are able to optimize correspondence over the entire 2D image rather than only along individual 1D scanlines, which enables information propagation from textured to untextured regions and thus solves the aperture problem to some extent. Global approaches use an energy framework to assess possible pixel correspondence of the entire image. What differentiates one global approach from another is mainly the way of finding the global minimum. Traditional approaches based on Markov Random Fields include simulated annealing [102], continuation methods [11] and mean field annealing [34]. Speed and robustness are the main concerns with these techniques.

More recently, graph cut based on the max-flow/min-cut theorem has been proposed to solve global optimization problems. It was first introduced to computer vision in 1989 [39] and popularized in the late 90's [14, 60, 108]. Again, a discrete disparity map is resolved in graph-based stereo rather than a continuous one. Graph cut is efficient compared to traditional global optimization methods and proved to produce good results.

There are two general schools of graph-based algorithms based on how they treat depth discontinuities. One school uses a linear cost function [60, 108] which enables reaching a global minimum but tends to produce over-blurred object boundaries. Others use a step

function to preserve discontinuity which makes global optimization NP-hard, *i.e.*, it is impossible to find a global solution efficiently. However, graph cut algorithms have been developed to compute a local minimum in a strong sense [12, 13, 69, 70, 71, 72]. This dissertation continues the idea of using discontinuity-preserving graph cut for energy minimization.

The main disadvantage of graph-based 3D algorithms is their limitation to computing discrete disparity values due to their inherently combinatorial nature. The recently emerging layered methods are an extension of graph cut that generate continuous disparity maps by introducing the concept of support maps. Each region in a support map identifies a patch of continuous surface. An optimum disparity map is computed by iterating among segmentation, surface fitting and graph cut [7, 80, 97, 142].

4.1.1 Nomenclature

The major symbols introduced in Chapter 4 and 5 are listed in Table 4.1.

Symbol	Explanation
γ	Smoothness energy coefficient
σ	Data energy coefficient
E	Energy
f	Labeling of the reference image
I	Pixel intensity
K	Static cue weight
n	Data energy coefficient
p, q	Pixels
P	Reference image
s	Source
$S\_threshold$	Static cue threshold
t	Sink
W	Fixed data energy weight

Table 4.1: Nomenclature used in Chapter 4 and 5.

4.1.2 Fundamentals of graph cut

In this section, we review the basics of graphs in the context of vision applications. Graph cut as a global optimization approach to solve pixel correspondance problem can be viewed as a two-step process. The first step is to represent the possible disparity assignments for all pixels with a weighted graph. The second step is to cut the graph, *i.e.*, find the optimum disparity assignment.

In order to understand the first step of building a graph, we need to begin with the energy minimization framework, and then learn how to assign energy terms to a graph.

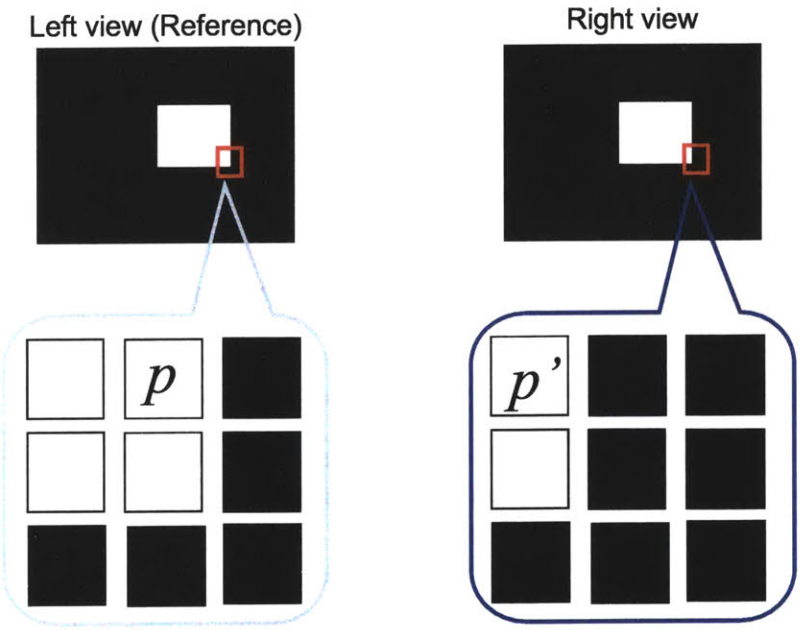

Figure 4-1: Image pair example with binary disparity values.

Let us look at a binary example as shown in Figure 4-1. Conventionally the left view is chosen as the reference image. In this example each pixel can only assume two disparity values: 0 or 1 *pixel*. Each possible disparity value is called a label. The foreground rectangle moves to the left by one pixel in the right view while the background remains stationary as illustrated in blowups. Pixel p in the reference image corresponds to p' in the right view image. Global energy is defined as

$$E(f) = \sum E_{data}(f) + \sum E_{smooth}(f) \tag{4.1}$$

where $f = \{f_p | p \in P\}$ is one possible labeling of the reference image P. Our objective is to find the optimum f that minimizes the total energy E.

The data energy term, $E_{data}(f)$, measures how well the image pair corresponds to each other under the disparity labeling f based on the assumption that corresponding pixel p in the reference image and $p + f_p$ in the second image should have similar intensities. Each pixel in P has a data penalty based on its label. The better the correspondence, the smaller the penalty. Total data energy is thus

$$\sum E_{data}(f) = \sum_{p \in P} D_p(f_p) \tag{4.2}$$

where $D(\cdot)$ is a non-negative data penalty function, for example,

$$D_p(f_p) = |I(p) - I'(p + f_p)| \tag{4.3}$$

The smoothness term, $E_{smooth}(f)$, encodes the smoothness assumption as a soft constraint based on the assumption that there is a high probability that neighboring pixels p and q have the same disparity. Thus the smoothness term penalizes neighbors that do not have the same disparity label:

$$\sum E_{smooth}(f) = \sum_{\{p,q\} \in N} V_{\{p,q\}}(f_p, f_q) \tag{4.4}$$

where N is the set of all neighboring pixels and $V(\cdot, \cdot)$ is a non-negative smoothness energy function, for example,

$$V(f_p, f_q) = \rho \cdot |f_p - f_q| \tag{4.5}$$

where ρ is some weight for disparity difference penalty. More choices used for data and smoothness terms will be discussed in Section 4.1.3.

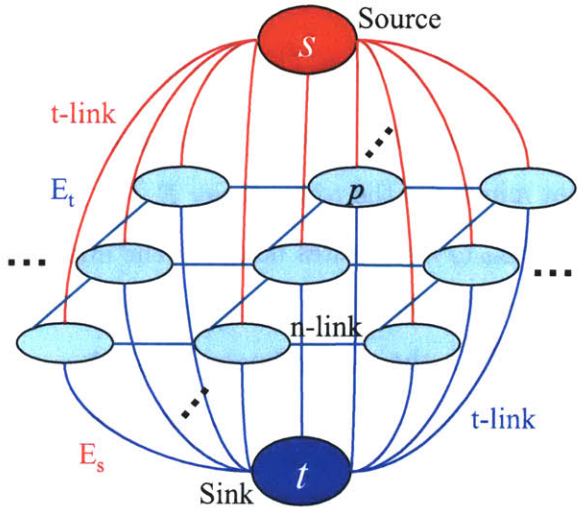

Figure 4-2: Example of a graph based on a image pair.

Next we incorporate the pixels in the reference image as well as energy terms into the data structure of a weighted graph. Figure 4-2 illustrates the graph built from the reference image in Figure 4-1. Only the nine blowup pixels are shown. A graph is comprised of a set of nodes and a set of links (can also be called edges or arms) that connect the nodes. There are two types of nodes: pixel or terminal. Each pixel from the reference image corresponds to one pixel node. There are two special terminal nodes which are called source, s, and sink, t, respectively. Terminal nodes correspond to the set of disparity labels that can be assigned to pixels. The definition of graph calls for quantized disparity values.

There are also two types of links in the graph: n-links and t-links. N-links connect neighboring pixel pairs. A four-neighbor convention is implemented in this dissertation. T-links connects pixel nodes with terminals. Before cutting, each pixel is connected to both the source and sink. All the links in the graph carry a weight or cost (capacity) determined by the energy terms. The weight for n-links corresponds to the smoothness penalty while t-links to the data penalty. The weight for the source link is computed as if the pixel is assigned a sink label. Similarly, the weight for the sink link is computed as if the pixel is assigned a source label. Note the reverse relationship here of weight assignments.

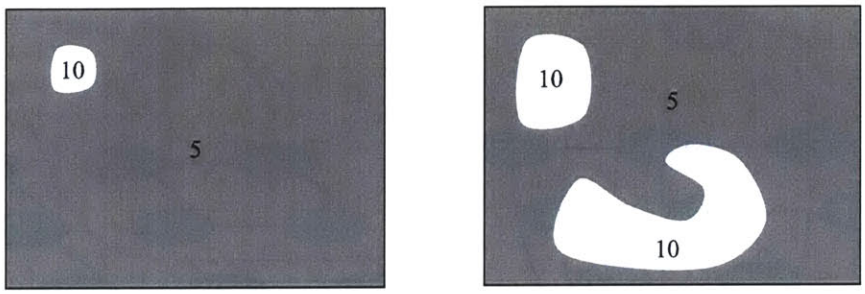

Figure 4-3: Example of α-expansion. Left: Initial labeling; Right: label 10 expands into other areas after expansion.

After explaining the energy model and composition of graph, we are finally ready to move to the second step, *i.e.*, to actually cut the graph. The expansion-move approach is one of the most efficient algorithm to cut a graph. As mentioned before, each disparity is called a label, and the label being solved is called an α-label. While solving for one disparity, this particular α-label will try to expand from its initial dominating regions. This process is called α-expansion. Figure 4-3 gives a simple example of α-expansion. The left image demonstrates the initial disparity map before expansion. Label 10 (*pixel*) occupies a small area with the rest belongs to label 5. After solving label 10, it may expand not only its original occupancy but also propagate to other areas.

In terms of the graph representation, an α-expansion is equivalent to cutting the graph. Figure 4-4 demonstrates one possible way of labeling for the nine pixels in Figure 4-1. The top left shows the graph before cutting and bottom left illustrates the initial disparity map. We suppose all nine pixels have an initial labeling f of 0 disparity for simplicity while in reality each of them can be initialized to 0 or 1 *pixel* in this simple binary example. If now we would like to solve for the disparity of 1 *pixel*, the source then corresponds to the α-label of 1 and the sink represents whatever the initial label is for each individual pixel. One possible scenario after expanding the α-label is that four pixel nodes are left connected to the source, while the rest to the sink as shown in the right column. As a result, the α-label expands to four pixels in the new labeling f'. A valid s-t cut must satisfy the following rule: each pixel node should be connected to at least one and only one terminal node. A connection

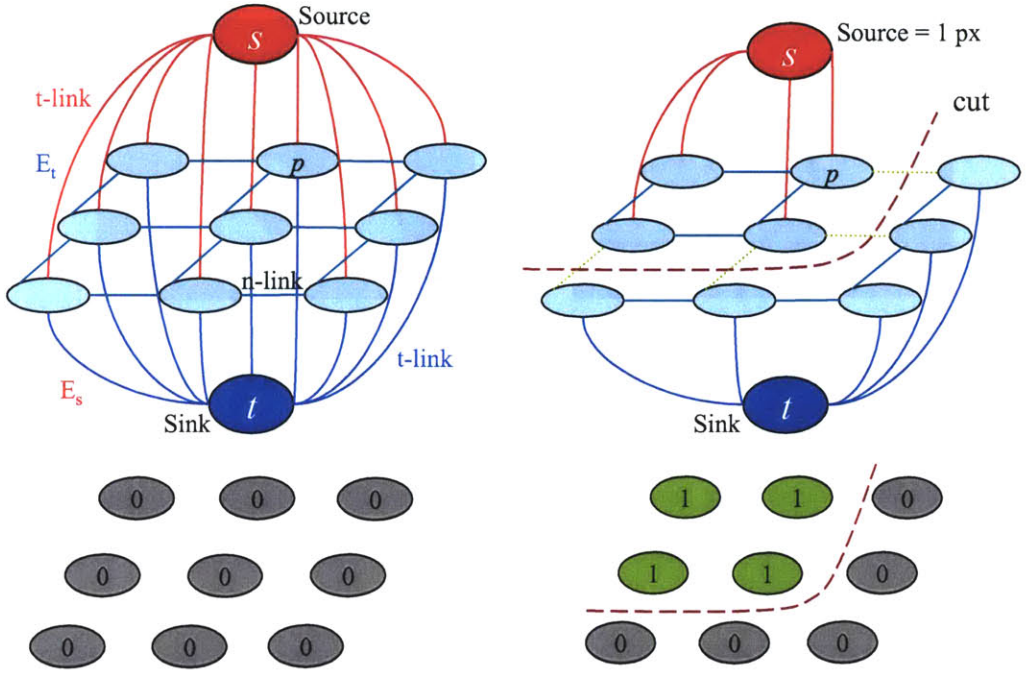

Figure 4-4: Graph cut example before and after expanding α-label 1.

between a pixel node and a terminal node after a cut means a disparity assignment. This way, the uniqueness assumption is satisfied by allowing one and only one disparity value per pixel node.

The cost of a cut is computed as the sum of the cost of all disconnected links, both n- and t-links. In the example of Figure 4-4, the total cost is the sum of the weights of four n-links, five disconnected t-links to the source and four t-links to the sink.

It is easy to prove that the cost of a cut is equal to the energy of a labeling f. A quick conceptual reasoning is as following: the total smoothness energy of the lower right labeling in Figure 4-4 equals to the sum of the cost of four n-links because all other n-links have a cost of zero; the total data energy equals to the sum of four E_s's of the four label-1 pixels and five E_t's of the five label-0 pixels, which are exactly the weights of the nine t-links cut.

Thus global energy minimization is reduced to the problem of finding the minimum cost cut, or min-cut. The famous Min-cut/Max-flow theorem states that the minimum cost cut of an initial graph is equivalent to the cut that allows the maximum flow from the source

to sink. Mathematicians have developed numerous max-flow algorithms over the past years. For big images with thousands to millions of pixels and a large disparity range, efficient cutting algorithms with polynomial complexity are a must. Graph-based methods rely on efficient algorithms [72] to closely approximate the minimum cost cut (or equivalently the maximum flow [23]) of a network graph. This thesis follows the max-flow implementation of [73] and uses it as standard comparison because of its popularity. As a generic module, the standard graph cut algorithm has been widely integrated with other modules such as surface segmentation and multi-image schemes to become much more sophisticated algorithms which usually require much longer computing time. By improving on the basic graph cut module, the proposed hybrid approach has the potential to be adopted by other higher level graph cut methods.

How to solve binary label problems has been explained so far. In real world applications, we usually need to work on multiple label optimization because various disparity values may be present. Graph cut processes all the possible labels (disparities) one at a time. For example, if the disparity range for the reference image is 0 to 15 pixels, a total of 16 α-labels should be cut. The source represents the current α-label while the sink whatever labeling from the previous cut. In each cut, a new graph is built and weights calculated based on the new source and new initial labeling. Then max-flow algorithm searches for a good cut. This new labeling becomes the initial condition for the next label being cut. A complete iteration traverses through every possible label in the disparity range. Total energy is usually computed at the end of each iteration. Several iterations are often required in practice to reach convergence of the total energy.

4.1.3 Standard energy model

The energy definition in the standard graph cut energy model is:

$$E(f) = \sum E_{data}(f) + \sum E_{smooth}(f) \tag{4.6}$$

where

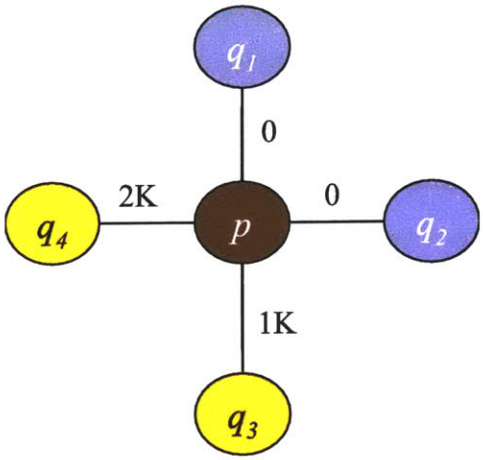

Figure 4-5: Example of assigning smoothness energy terms to n-links.

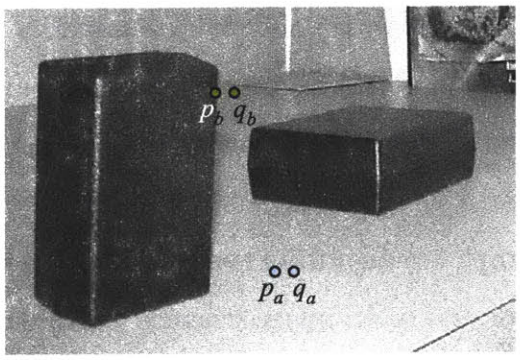

Figure 4-6: Observation for applying static cue.

$$E_{data}(f_p) = |I(p) - I'(p + f_p)|^n \tag{4.7}$$

$$E_{smooth}(f_p, f_q) = \begin{cases} 0 & f_p = f_q \\ K & |I_p - I_q| > S\_threshold \text{ and } f_p \neq f_q \\ \gamma K & |I_p - I_q| \leq S\_threshold \text{ and } f_p \neq f_q \end{cases} \tag{4.8}$$

By summing up all data energy terms for every pixel and smoothness energy for all neighbors, we have the total energy for one possible labeling f. Data energy of each pixel is calculated by comparing the intensity difference between the two corresponding pixels defined by f_p. n is a positive integer usually set to 1 or 2. When calculating data energy term, sub-pixel intensity comparison is implemented in preference to integer pixel in order to reduce the pixelization noise [6].

Each smoothness energy term is computed by checking whether neighbors have the same disparity. Figure 4-5 demonstrates how to assign cost for the four n-links of pixel p. Suppose pixels q_1 and q_2 have the same label as p. Thus their smoothness penalty is zero. When neighboring pixels have different labels such the case of q_3 and q_4, static cue applies to determine their smoothness penalty. Static cue is based on the observation that adjacent

pixels with similar intensity values are likely to belong to the same object and consequently the same disparity, such as the two neighbors p_a and q_a in Figure 4-6. A larger penalty of γK discourages p_a and q_a to have different labels in Equation 4.8. $\gamma > 1$ is a pre-defined weight, *e.g.* 2 or 4. When the intensity values of two adjacent pixels are quite different, such as the two neighbors p_b and q_b, they are likely to belong to different objects. A smaller smoothness penalty gives graph cut algorithm more flexibility to assign different labels to p_b and q_b. K is often set to about 10 and $S\_threshold$ to 5 grayscales.

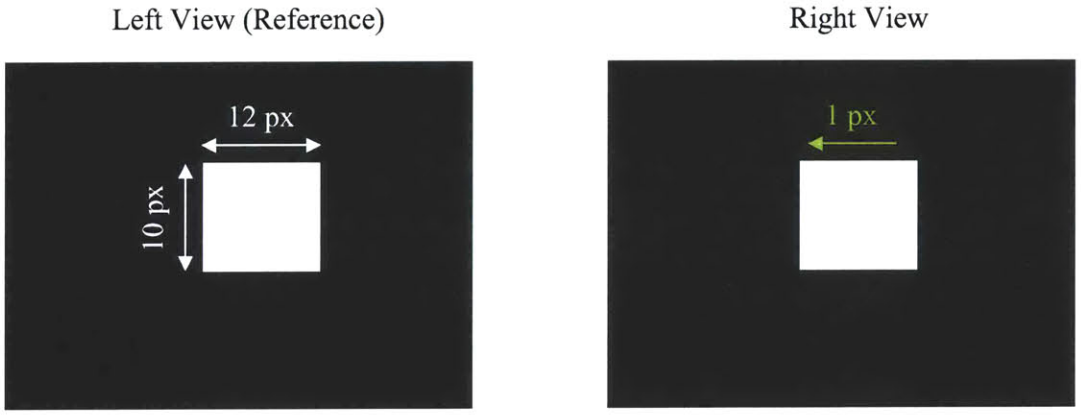

Figure 4-7: Static cue example. A white block of size 12 × 10 *pixel* is shift to the left by 1 *pixel.*

Next let us look at a simple example of how static cue helps propagate useful disparity information into textureless regions. A white block of size 12×10 *pixel* and intensity 255 grayscales is shifted to the left by one pixel in Figure 4-7. Suppose the background has an intensity value of 0 and there is no noise in the two intensity images. In this case, energy minimization is determined by the smoothness terms. The two most possible solutions where $\sum E_{data}$ are zero are illustrated in Figure 4-8. Dark color represents a label of 0 *pixel*, while bright color 1 *pixel.* The dashed area indicates the foreground object's location in the reference image. Each white strip has a width of one pixel and height of 10 pixels in the left labeling. The foreground object is assigned a width of 13 pixels and height of 10 pixels in the right labeling. Intuitively, we know the right labeling is closer to ground truth. However,

graph cut can only choose the labeling with the minimum total energy.

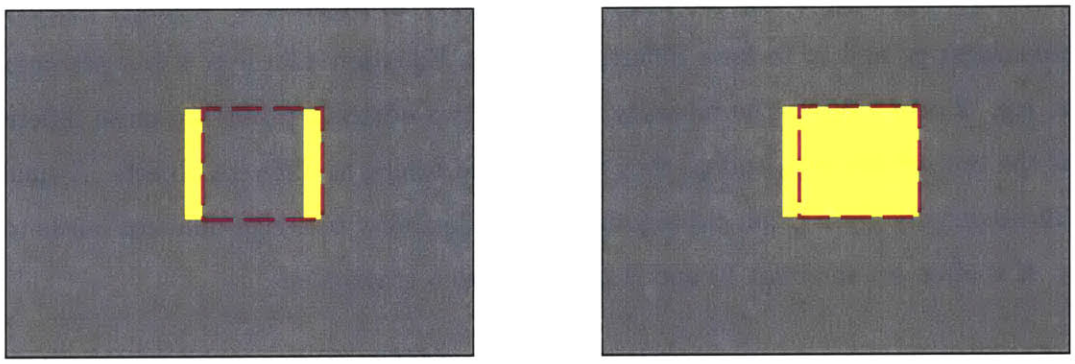

Figure 4-8: Two possible labelings for the static cue example. Dark color represents a label of 0 *pixel*, while bright color 1 *pixel*. Left: wrong solution. Right: more accurate solution.

If there is no static cue in the smoothness energy term in Equation 4.8, then its definition is reduced to, for example

$$E_{smooth}(f_p, f_q) = \begin{cases} 0 & f_p = f_q \\ 2K & f_p \neq f_q \end{cases} \tag{4.9}$$

where $K = 10$. We can compute the total energy as: $\sum E_{smooth} = 20 \times 44 = 880$ for the left labeling; and $\sum E_{smooth} = 20 \times 48 = 960$ for right labeling. Here 44 and 48 are the number of neighbors with different labels in each respective labelings. Graph cut algorithm will prefer the wrong labeling over the right one after comparing total energies when static cue is not introduced.

If there is static cue in smoothness energy terms as in Equation 4.8, then the total energies are computed as: $\sum E_{smooth} = 10 \times 12 + 20 \times 10 + 10 \times 10 + 20 \times 12 = 660$ for the left labeling; and $\sum E_{smooth} = 10 \times 34 + 20 \times 12 = 580$ for the right labeling. This time, the more accurate solution will be chosen after cutting label 1.

Note that the occlusion problem around the left boundary is not solved by this standard energy model. The perfect solution should be exactly the size of the dashed area. However, the data energy is no longer zero in the perfect labeling due to occlusion, but rather $\sum E_{data} = 255 \times 10 = 2550$ if we set $n = 1$ in Equation 4.7. The smoothness term becomes insignificant

compared to the data term at $\sum E_{smooth} = 10 \times 44 = 440$. Total energy for the perfect labeling is 2990, which is much larger than 580 of the right labeling in Figure 4-8. As a result, graph cut will choose the latter one as final output. More sophisticated energy models that take occlusion into consideration have been proposed in literature [60, 70, 79, 142].

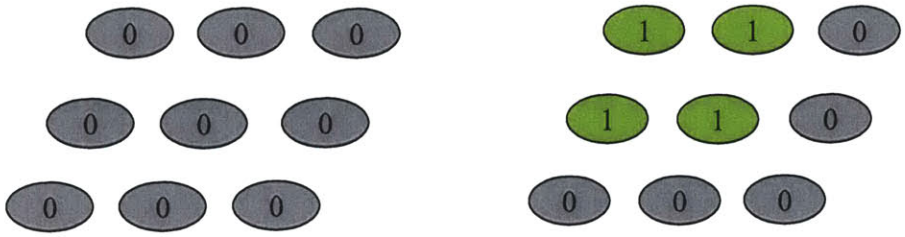

Figure 4-9: Two possible labelings for the binary example in Figure 4-1. Left: wrong; Right: correct.

Let us go back to the binary example in Figure 4-1 and see how the nine pixels can be correctly cut following the standard energy model. Only two possible labelings are shown in Figure 4-9 out of the $9^2 = 81$ potential labelings. Total energies are computed in Table 4.2 for both wrong and correct labeling. Actually total energies of the other 79 possible disparity maps are all larger than 40. Graph cut will prefer the correct labeling over wrong ones based on energy minimization.

Energy	Wrong f	Correct f
$\sum E_{data}$	$255 \times 2 = 510$	0
$\sum E_{smooth}$	0	$10 \times 4 = 40$
E_{total}	510	40

Table 4.2: Energy calculation of the two labelings in Figure 4-9.

4.2 Difficulties of standard energy model

Graph-based methods have been gaining momentum in computer vision in recently years due to their capability to generate a dense disparity map robustly. However, they have their own issues.

4.2.1 Initial conditions

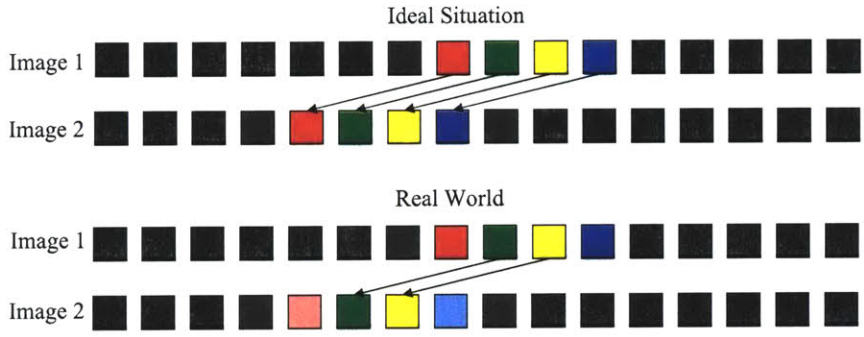

Figure 4-10: 1D image pairs in ideal situation and real world.

There is an inherent flaw with the standard energy model, which is troublesome for all correspondence based 3D algorithms. We need to ask this question: are data terms a reliable test for corresponding pixels? Prevailing systematic or random image noises, which are everywhere in real images, make E_{data} unreliable. For example, the upper one-dimensional image pair in Figure 4-10 only exists in simulation. The black background stays put between frames. The four pseudo-colored pixels shift to the left by three pixels and do not change their intensity values. However, in real world, image noise can be dominant. For example, noise may come from sensor dark current, flare or illumination variation from frame to frame. In real world scenarios, the red and blue pixels may change their intensity values significantly as shown in the lower example. As a result, only green and yellow pixels may be properly resolved to have a disparity of three pixels under the standard energy model.

Here is a real example of how image noise is misleading for standard graph cut. Figure 4-11 illustrates the reference left view of the "Room" pair as well as blowups of both views. The first frame is rougher while the second one smoother. The area under scrutiny sits across the conference room wall and door frame. Hand calibrated ground truth indicates that the background wall has a disparity around 2 *pixel* and the door frame 3 *pixel*, while the closer the object is to the camera the larger the disparity. The foreground table has a disparity of around 15 *pixel*. The two solid-line oval regions under a white spot in the blowups should

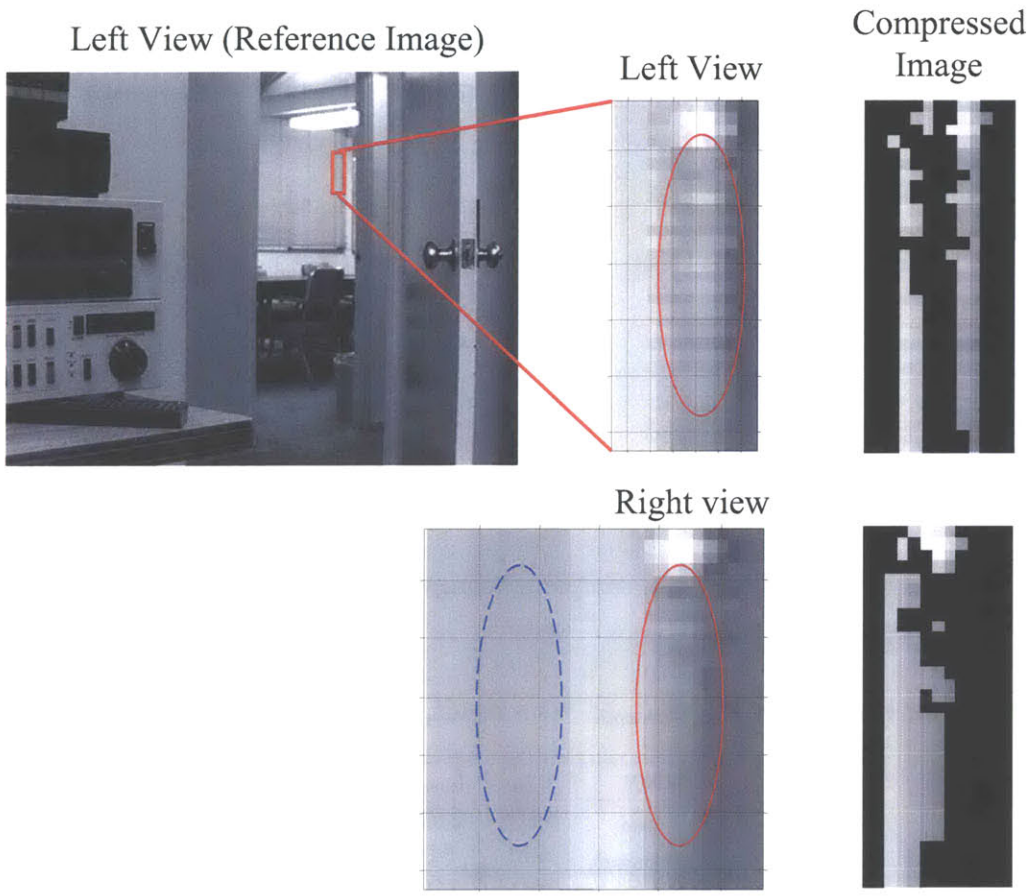

Figure 4-11: Blowup of the "Room" image pair.

correspond to each other. However there is obviously significant noise between frames: the reference image is more varied and the second image more uniform. Consequently, the two solid-line oval regions do not have a good match based on data energy calculations. Instead, the oval region in the reference image finds a better match with the dashed-line area 15 pixels to the left in the second image as explained in Figure 4-12.

Two sample labelings are presented in Figure 4-12. For each labeling, the data energy of each pixel is also shown as a brightness image. The good labeling f_{good} is more reasonable because only labels 2 and 3 are included. The bad labeling f_{wrong} has a region of label 15 in the center. We can notice that data energy distribution of f_{good} looks brighter overall and thus a larger $\sum E_{data}$ than f_{wrong}. Exact energy calculations are given in Table 4.3. $\sum E_{data}$

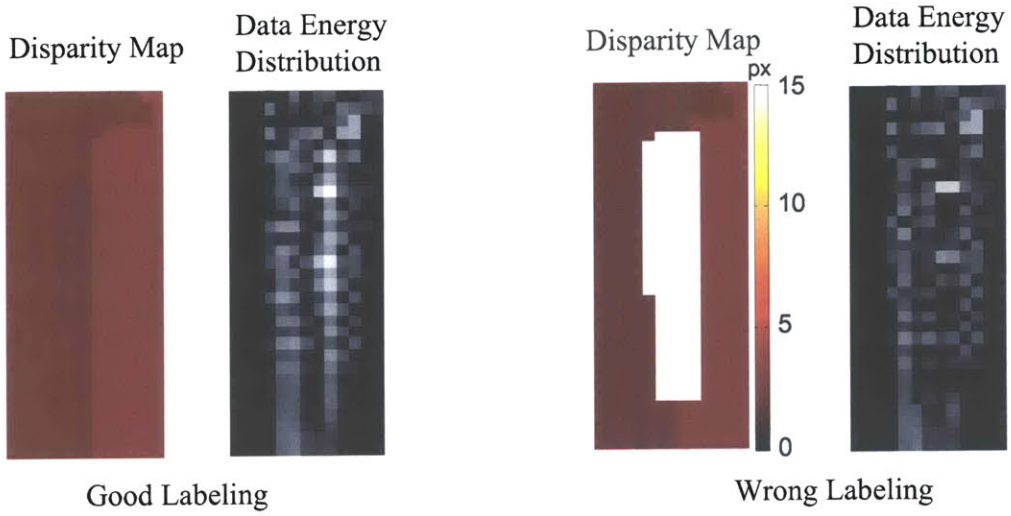

Figure 4-12: Energy distribution of two sample labelings using standard energy model. Left: f_{good}; Right: f_{wrong}.

is dominant in our example over $\sum E_{smooth}$. As a result, f_{wrong} is preferred by graph cut.

Energy	f_{good}	f_{wrong}
$\sum E_{data}$	3418	2693
$\sum E_{smooth}$	490	810
E_{total}	3908	3503

Table 4.3: Energy calculation of the two labelings in Figure 4-12 using the standard energy model.

What if we know in advance the ground truth of some pixels? Will their labels remain the same after cutting based on noisy intensity values? In the example of Figure 4-13, suppose we are going to cut disparity 15 *pixel*. The source α-label is then 15. Before the cut, we have the left graph. All possible t- and n-links are connected and weights are assigned. Each pixel has an initial condition. Suppose we know *a priori* that pixel p in Figure 4-13 should have a disparity of 2 *pixel*. Its two t-links have the following E_{data} values, which are correct based on noisy intensity comparisons: if the pixel has the source label of 15, the data penalty is 10; if it has the sink label of two, the data energy is 40. In other words, the penalty is larger if this pixel belongs to sink. Graph cut will try to find the minimum cut, *i.e.* to cut all

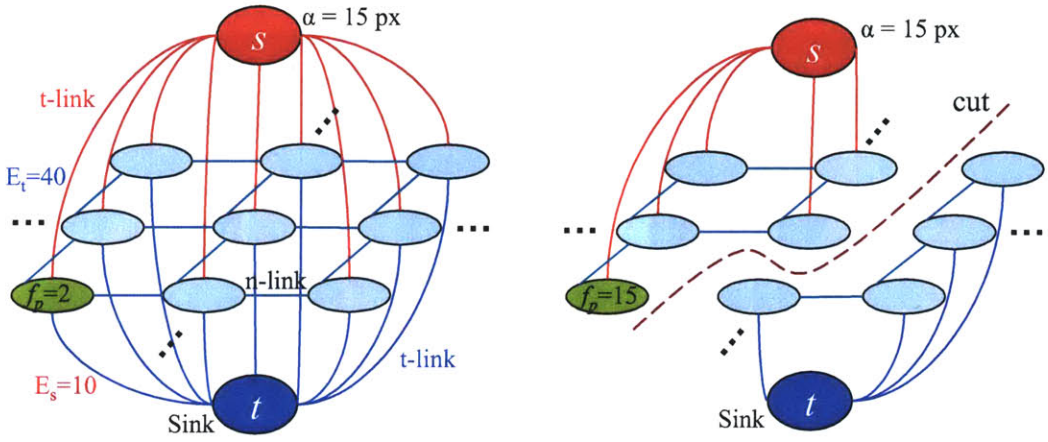

Figure 4-13: Example of cutting α-label 15 using standard energy model.

the links that have the minimum total cost. Then there is a high probability that the t-link connected to the sink with a much smaller weight will be cut leaving p a wrong label of 15 *pixel*. This is where most gross errors in graph cut algorithms come from.

It is a desirable property that graph cut can integrate reliable prior information and does not modify their values after cutting. However, such integration is not an easy task. As a first instinct, placing prior information into initial conditions seems to be a viable and simple solution. We know that in the continuous domain, a better initial condition often leads to faster convergence and global rather than local optima. However unlike continuous minimizations, graph cut as a combinatorial optimization approach is insensitive to initial conditions. For example, the hand calibrated ground truth for the "Tsukuba" image benchmark is provided in Figure 3-13 where lighter grayscale values indicate larger disparities. Initial conditions of graph cut can be arbitrarily set to any labels in the predefined disparity range, *e.g.*, all pixels may have a disparity of 0 *pixel*, or ground truth. The results of "Tsukuba" based on different initial conditions using the standard energy model are illustrated in Figure 4-14. A total of 16 labels in the decreasing order from 15 to 0 *pixel* are cut in each scenario. The ground truth is deteriorated right after cutting label 15. The final result based on ground truth is hardly any better than when the initial condition is set to all zeros.

It will be very preferable for graph cut to take advantage of prior information, since often

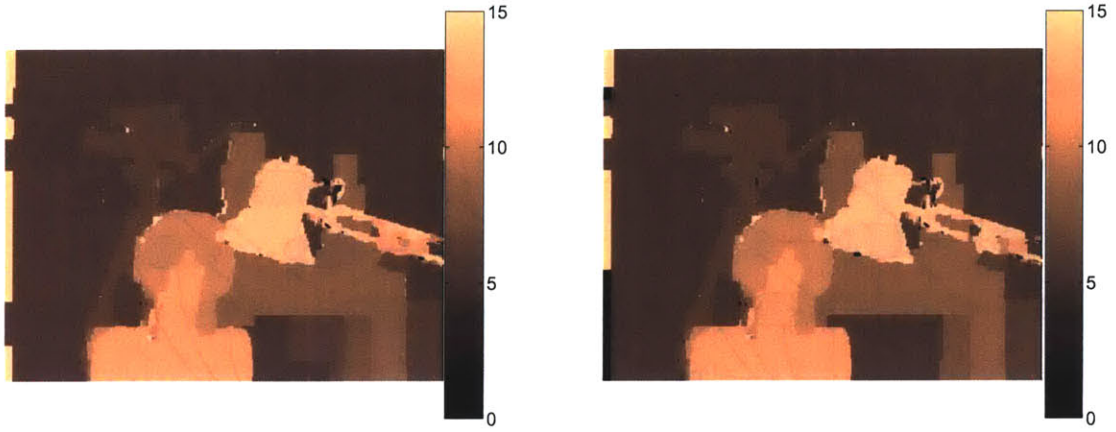

Figure 4-14: Output from "Tsukuba" based on standard graph cut energy model. Left: initial condition is 0 *pixel* disparity for all image pixels. Right: initial condition is ground truth.

the user may have some manual input or know the position of some tracking targets. We need to find out other ways to integrate prior information into graph cut energy model than mere initial conditions.

4.2.2 Speed

Another disadvantage of standard graph cut is that it requires several iterations (usually three) to reach convergence. This practice is not preferable to real-time applications. One interaction on a 500×500 *pixel* image pair often takes seconds. Most of the fast graph cut algorithms nowadays do not traverse through all possible labelings, resulting in a local minimum within a factor to the global minimum. Because the process of cutting one α-label is only an approximate solution, it is a common practice to randomize the order of cutting among all the labels in a complete iteration. How to further reduce the computational time of one interaction for a fixed number of pixels and labels is a working progress among algorithm developers. This thesis focuses on reducing the number of iterations while at least maintaining accuracy.

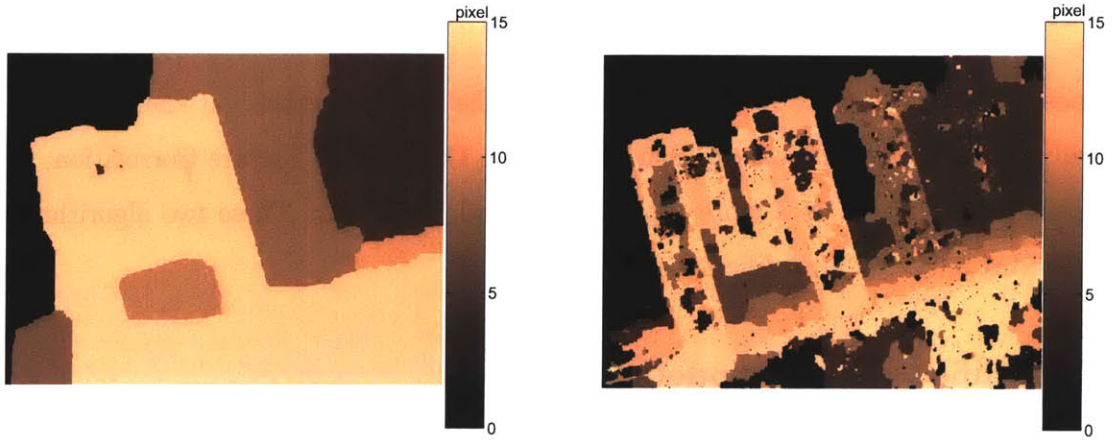

Figure 4-15: Output from "MIT" pair based on standard graph cut energy model. Left: $n = 1$. Right: $n = 2$.

4.2.3 Parameter sensitivity

The third drawback of standard energy model is its sensitivity to the parameter settings in the model, such as n in Equation 4.7, K and $S\_threshold$ in Equation 4.8. Figure 4-15 demonstrates how the standard graph cut result of the "MIT" pair in Figure 3-12 is influenced by n. Other parameter settings are the same: $K = 10$, $\gamma = 2$ and $S\_threshold = 5$. The "MIT" pair is especially difficult for dense correspondence methods because it has large untextured areas. Such kind of images are particularly sensitive to the parameter n when K and γ are fixed, resulting in dramatic difference in the two graph cut outputs in Figure 4-15. $n = 2$ places a much larger weight on E_{data} than $n = 1$. Consequently, the role of E_{smooth} to enforce smoothness is diminished. There are much more random dots and patches in the disparity map when $n = 2$ due to ambiguity and image noise. In contrast, when $n = 1$ the smoothness term successfully propagate good disparities into untextured even though it has a tendency to over- smooth under the current parameter settings. Optimal settings can be found by carefully balancing all the parameters in Equation 4.7 and Equation 4.8.

It is impossible that a single set of energy model parameters would suit any types of images. Our goal is to make graph cut more robust by curtailing the output degradation when shifting from the optimal parameter settings.

4.3 Hybrid approach

The proposed solution to the above three issues with the standard graph cut energy model is a hybrid approach. A much less expensive algorithm, Compressed Feature Correlation, is used to stabilize and speed up a more expensive method, graph cut. These two algorithms are integrated together by a modified energy model.

4.3.1 Modified energy model

As mentioned in Section 4.2.1, unreliable data energy computation due to image noise is one of the main sources of gross errors in graph cut. Suppose we have accurate disparity estimations for a subset of pixels $Q \in P$. And $P - Q$ stands for the subset of pixels without any prior information. For pixels with priors, their data energy values can be fixed instead of using noisy intensity comparisons in order to maintain their original labels. Prior information is introduced into a modified energy model through fixed data energy cost while smoothness energy definition remains the same as in Equation 4.8.

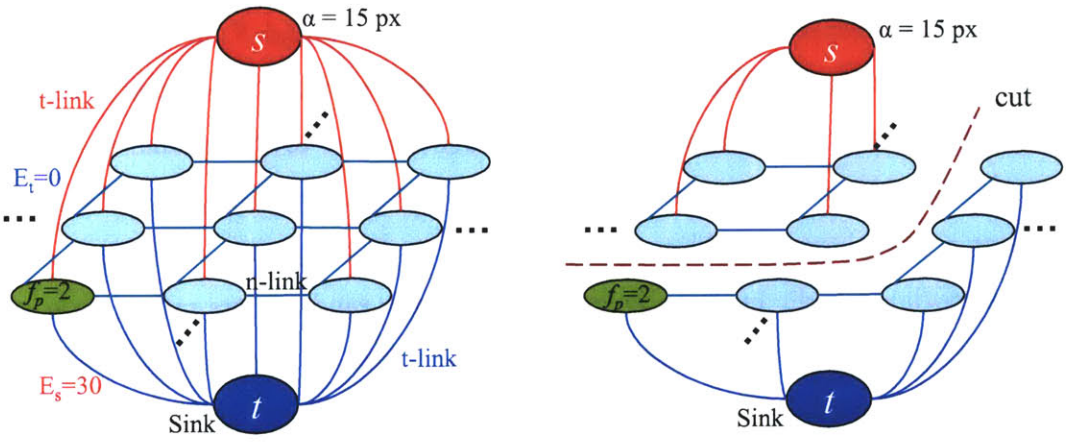

Figure 4-16: Example of cutting α-label 15 using priored energy model.

Figure 4-16 gives an example of how data energy terms are computed using a priored energy model when cutting α-label 15. Pixel p is known to have a disparity around 2 *pixel*. However, it would be labeled 15 by graph cut using standard energy model as explained in

Figure4-13. To prevent a label change, p should belong to the sink and remain its initial condition of 2 *pixel*, which means that the t-link connected to s should be cut and leave the one to t connected. Thus the weight of the former link can be forced to take the smallest possible weight of 0, and the latter link a large weight, *e.g.* 30. This way, p has a high probability of keeping its labeling after cutting. This is our proposed way of integrating prior information into the standard graph cut model and make it relevant to initial conditions. The hybrid approach may be compared in analogy to continuous optimization. Instead of setting boundary conditions on the reference image boundary as the practice in the continuous domain, discrete "boundary conditions" are added in the middle of surface. The optimization outcome of some specific spots are constrained to preset values. Consequently, neighboring pixels are also influenced by the priors due to the smoothness assumption.

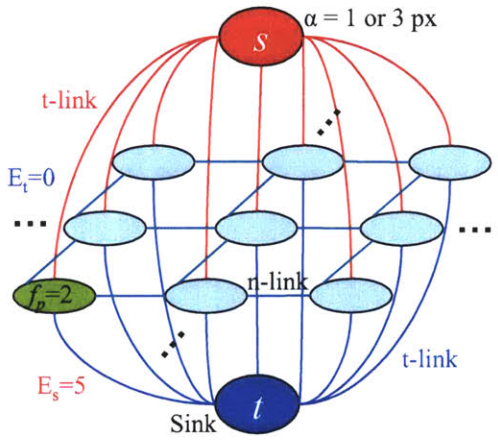

Figure 4-17: Example of cutting α-label 3 using priored energy model.

Because prior disparities may have ± 1 *pixel* accuracy, one variation is introduced into the modified energy model. The label of $p \in Q$ is allowed to convert to ± 1 of its predetermined disparity value by reducing its shifting cost as illustrated in Figure 4-17. When cutting α-labels 1 or 3, the t-link connected to the sink has a much smaller fixed cost of 5. This way it could potentially be cut and p converts to a new label if smoothness energy is minimized at the same time.

In summary, prior information is integrated into the standard graph cut model by using

the following modified energy model, where smoothness energy definition is the same as Equation 4.8.

$$E(f) = \sum E_{data}(f) + \sum E_{smooth}(f) \tag{4.10}$$

$$E_{smooth}(f_p, f_q) = \begin{cases} 0 & f_p = f_q \\ K & |I_p - I_q| > S\_threshold \text{ and } f_p \neq f_q \\ \gamma K & |I_p - I_q| \leq S\_threshold \text{ and } f_p \neq f_q \end{cases} \tag{4.11}$$

E_{data} is defined in Table 4.4. W is the fixed data cost for switching to $\alpha = f_p \pm 1$ and is often set to 5. Coefficient $\sigma > 1$ can be set to 6-10.

$E_{data}(f_p)$	$p \in P - Q$	$p \in Q$		
		$\alpha = f_p$	$\alpha = f_p \pm 1$	else
E_s	$\|I(p) - I'(p + f_s)\|^n$	$\|I(p) - I'(p + \alpha)\|^n$	W	σW
E_t	$\|I(p) - I'(p + f_t)\|^n$	$\|I(p) - I'(p + \alpha)\|^n$	0	0

Table 4.4: Data energy definition in the modified energy model with priors.

[72] specifies what energy models can be minimized using graph cut. It is easy to prove that the above modified energy model can be solved using graph cut because the form of $D_p(\cdot)$ does not matter and $V_{\{p,q\}}(\cdot,\cdot)$ is a metric.

Let us re-visit the real example in Figure 4-11 to see how the proposed hybrid approach can successfully reduce the damage of image deterioration. The two compressed views are used in Compressed Feature Correlation. As a result, all the non-black pixels in the reference view have an estimated disparity of 2 *pixel*. There are numerous potential labelings after cutting label 15 and the one with the minimum total energy will be chosen as output. Two sample labelings are illustrated in Figure 4-18, whose energies are listed in Table 4.5. We can see that this time f_{wrong} has a larger total data energy than f_{good} mainly due to the heavy penalties given to the priored pixels who are mistakenly permitted to change labels. Combined with $\sum E_{smooth}$, graph cut algorithm will prefer f_{good} to f_{wrong}. With the hybrid approach, some erroneous outputs can be prevented and overall accuracy improved.

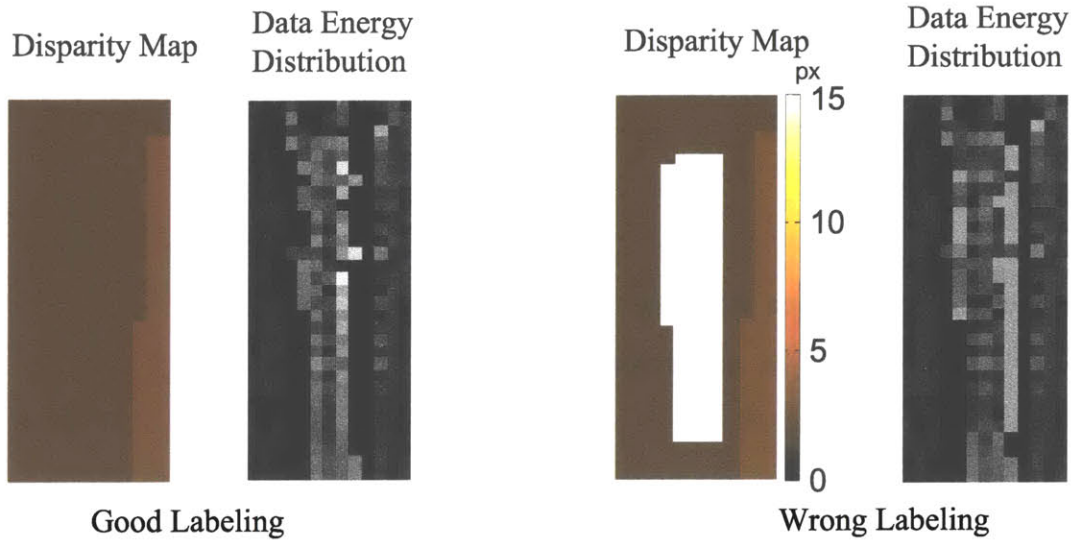

Figure 4-18: Energy distribution of two sample labelings using priored energy model. Left: f_{good}; Right: f_{wrong}.

Energy	f_{good}	f_{wrong}
$\sum E_{data}$	2252	2435
$\sum E_{smooth}$	370	970
E_{total}	2622	3405

Table 4.5: Energy calculation of the two labelings in Figure 4-18 using the priored energy model.

Sample pseudo-code implementation for building the graph and assigning data costs when cutting an α-label is presented as following:

```
1 void BuildGraph(void) {
2 for every pixel p in the reference image
3   get f_p from last cutting or initial condition
4   if p has no prior
5     if f_p = α
6       add constant data penalty E_s = E_l
7     else (f_p needs to be minimized)
8       calculate E_s and E_t
9     end if
10  else (p has prior)
11    if f_p = α
12      add constant data penalty E_s = E_t
13    else if α = f_p ± 1
14      calculate E_s and E_t
15    else (f_p would not change)
16      calculate E_s and E_t
17    end if
18  end if priors
19 end for p
20 }
```

4.3.2 Discussions

A key step in the hybrid approach is obtaining reliable prior disparity information. Other than the compressed feature correlation approach proposed in this thesis, there might be other ways to get priors from manual user input or feature-based 3D tracking methods. For example, the user may know that all the blue pixels should represent the sky and therefore have an disparity of 0 *pixel* between frames. Or the user may have disparity information for the markers attached to a patient's body tracked by ultrasound or laser devices.

The two steps of the proposed hybrid approach may be viewed as two independent modules. The output of compressed feature correlation can be used in applications other than dense disparity map generation, such as segmentation and tracking. The modified graph cut energy model can take priors from any other reliable sources.

Chapter 5

Performance evaluation of the hybrid approach

In this chapter, experiments are performed on a number of stereo pairs to compare the performance of the proposed hybrid approach and standard graph cut. Results show significant improvement in speed and accuracy using the hybrid approach. It's limitations are also discussed in Section 5.3.

The hybrid approach first runs Compressed Feature Correlation and then one iteration of graph cut with the modified energy model (Equation 4.11 and Table 4.4). Standard graph cut runs for three iterations because one iteration alone usually renders poor results. Only grayscale datasets are used, which run faster but might have slightly worse accuracy than color images. Parameter settings shared by the two algorithms such as n, K and $S\_threshold$ are exactly the same. Resulting disparity maps for qualitative experiments are illustrated as intensity images with disparity values assigned for all pixels. Error statistics are computed for datasets with ground truth. Running times are obtained on a laptop with a 2GHz Pentium 4 processor. Occlusions are not explicitly modeled or marked in this thesis.

5.1 Qualitative results on real images

Figure 5-1 compares the results of both hybrid and standard graph cut on the "Room" image pair. The equipment and table in the foreground have a large disparity around 13 *pixel*. The far conference room background has a small disparity around 2 *pixel*. Other objects have a disparity in between. Using the hybrid method, the disparities of some strong features are first calculated with compressed feature correlation as shown in Figure 3-7. This process is very fast using only 16 *ms* for the 630×480 *pixel* image pair. Then, priored graph cut with modified energy model is used to find the full field disparity map ($n = 1$, $K = 10$, $\gamma = 2$, $S\_threshold = 7$). Only one iteration is performed with the hybrid approach.

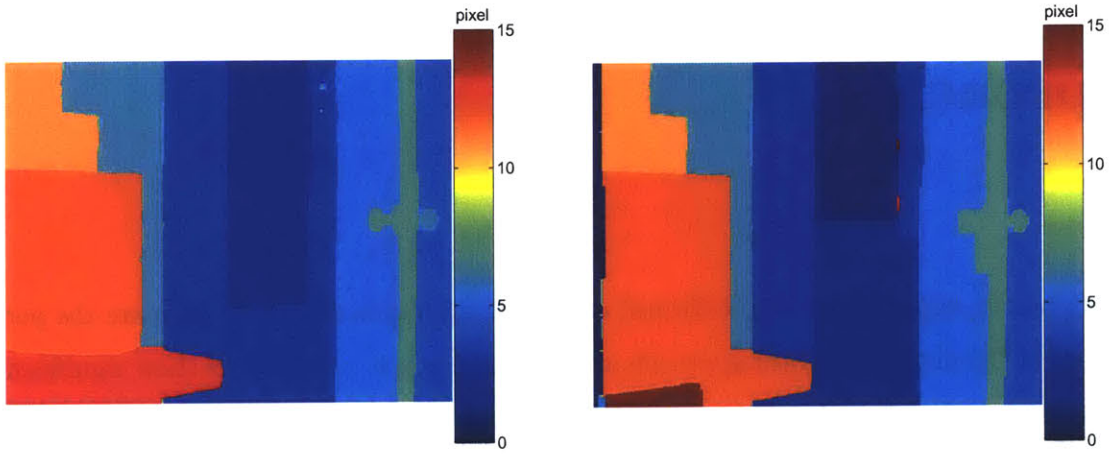

Figure 5-1: Graph cut output from "Room" image pair. Left: modified energy model with priors. Right: standard energy model.

An important question to ask is whether the coupling of two algorithms affects the accuracy of the final output. Figure 5-1 demonstrates that the hybrid approach produces a better result than standard graph cut with less than one third of the time. Three iterations and 29 seconds are necessary to get a good estimate using standard graph cut, while only 8 seconds for the hybrid method. The addition of Compressed Feature Correlation hardly takes any time compared to graph cut algorithm. By comparing the two results in Figure 5-1, the hybrid method has a comparable or even better accuracy. For example, the door knobs are correctly separated from the door; the table is separated from other equipments; and

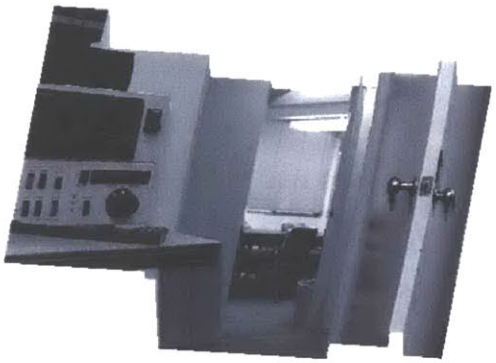

Figure 5-2: Texture-mapped rendering of "Room" output using modified energy model with priors.

the background conference room is recognized from its surrounding walls, door frame and foreground floor. A texture-mapped disparity rendering of the hybrid result is presented in Figure 5-2.

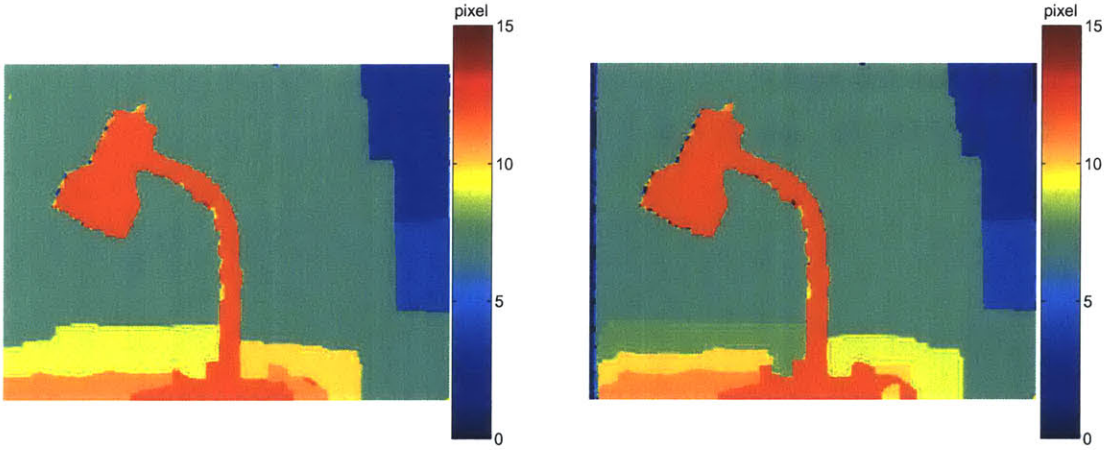

Figure 5-3: Graph cut output from "Lamp" image pair. Left: modified energy model with priors. Right: standard energy model.

Figure 5-3 shows the graph cut results on the "Lamp" image pair with or without priors ($n = 2$, $K = 10$, $\gamma = 2$, $S\_threshold = 10$). Compared to the standard graph cut, the output quality of the hybrid method is comparable if not better while the computing time is less than one third.

Figure 5-4 and Figure 5-7 present two additional datasets as well as their Compressed

Feature Correlation outputs. Both of them are taken in dark rooms with directional lighting. The subject in the "Reef" pair is a coral reef ornament about two inches high. Image size is 640×512 *pixel* and the disparity range is from 0 to 3 pixels. The "Teeth" pair with an image size of 768×1024 *pixel* studies two real teeth and its disparity range is from 0 to 65 pixels. Such uniform black background datasets provide additional prior knowledge which helps increase speed and robustness. All pixels with intensities lower than a threshold should belong to the background and thus have a prior label of 0 *pixel*. When initial conditions are set to zero, energy terms for background pixels might be set as follows:

```
1 void BuildGraph(void) {
2 ...
3 if I(p) < 20
4    if α = 0
5       add constant data penalty E_s = E_t
6    else (f_p remains zero)
7       E_s = INFINITY and E_t = 0
8    end if
9 else if
10 ...
11 }
```

where INFINITY is often chosen to be larger than 1000.

There is significant illumination variation between the two frames of "Reef", especially in the right part. Compressed Feature Correlation successfully calculates disparity estimates for major features as illustrated in Figure 5-5 and these prior information significantly stabilizes and speeds up graph cut. Figure 5-6 shows the graph cut results on the "Reef" image pair with or without priors ($n = 1$, $K = 10$, $\gamma = 4$, $S\_threshold = 5$). Standard graph cut completely fails in the noisy regions even with three iterations that take 22 seconds, while priored graph cut takes less than 1.5 seconds and successfully produces correct disparity

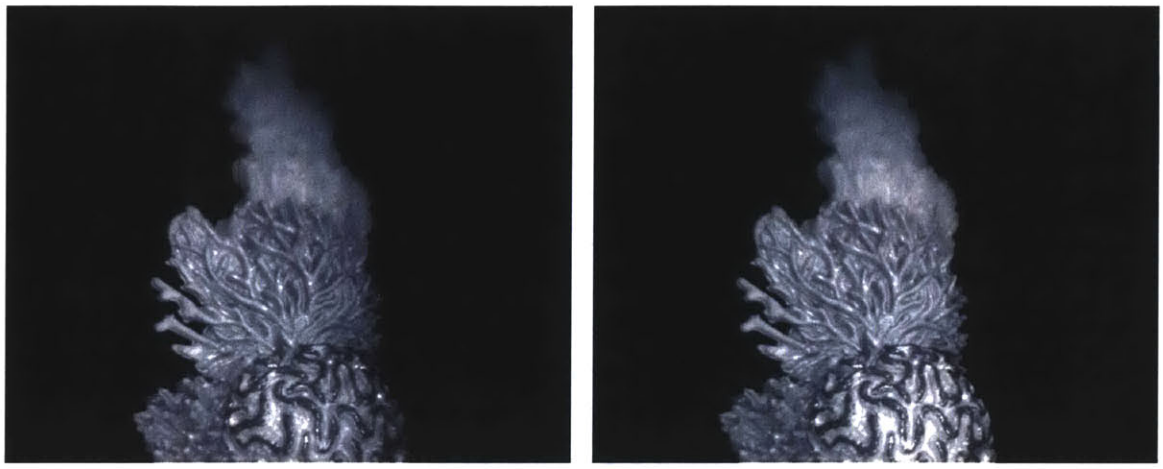

Figure 5-4: Left: left view reference image of "Reef". Right: right view.

estimations in the noisy region .

The "Teeth" pair is a very challenging dataset for pixel correspondence algorithms. The object surface is overall smooth and there is hardly any surface texture. Shiny speckles change their shape and brightness at different viewpoints. Sensor dark noise is predominant. What makes the situation worse is that the disparity range is especially large and the chance of ambiguous mismatch is greatly increased. Compressed correlation only picks out a moderate amount of prior pixels as illustrated in Figure 5-7.

Despite all the difficulties, the modified energy model still does a better job than the standard one. Figure 5-8 shows the graph cut results on the "Teeth" image pair with or without priors ($n = 2$, $K = 10$, $\gamma = 2$, $S\_threshold = 5$). The dark background is correctly separated from the objects using our algorithm and the disparity map has much less gross errors. The small black holes due to image noise could be easily fixed by post-processing such as median filtering. A texture mapping rendering of the disparity map generated by the hybrid approach is presented in Figure 5-9. Three iterations of standard graph cut take six minutes while one iteration with priors only takes 23 seconds. Again, the computation time of compressed feature correlation is negligible at 0.09 second.

The "Box" scene is another difficult image type with large untextured regions. Figure 5-10 shows the graph cut results on the "Box" image pair with or without priors ($n = 1$,

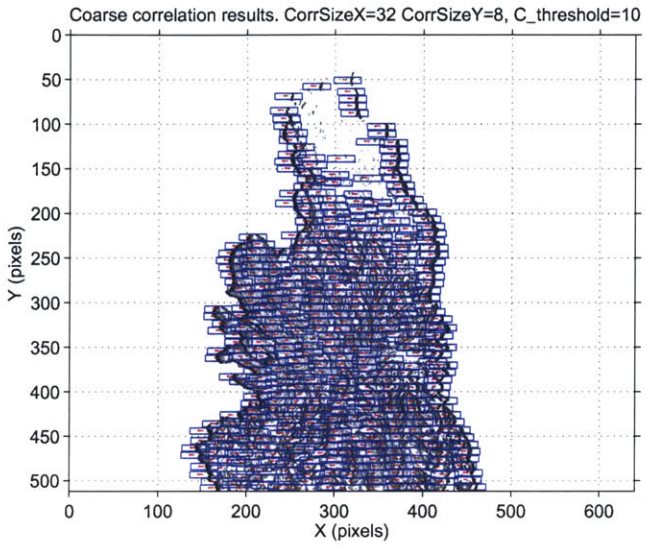

Figure 5-5: Compressed Feature Correlation output of "Reef".

$K = 10$, $\gamma = 2$, $S\_threshold = 18$). Modified energy model is better at resolving the table floor and separating the two boxes from background. In addition there is no black holes of disparity 0 *pixel* as in the standard graph cut output.

5.2 Quantitative results on benchmark images

Finally the proposed hybrid approach is tested on benchmark images with ground truth. For all the results presented in Section 5.2, the energy model parameters are set to the values in Equation 5.1 and Table 5.1 if not mentioned otherwise.

$$E_{smooth}(f_p, f_q) = \begin{cases} 0 & f_p = f_q \\ 10 & |I_p - I_q| > 5 \text{ and } f_p \neq f_q \\ 20 & |I_p - I_q| \leq 5 \text{ and } f_p \neq f_q \end{cases} \tag{5.1}$$

Compared to some qualitative datasets in Section 5.1, the benchmark images have ample surface texture to suit the needs of most 3D algorithms. The white regions defined as untextured areas in the error masks are scattered and small in scale. The "Map" pair is especially densely textured and has no untextured regions.

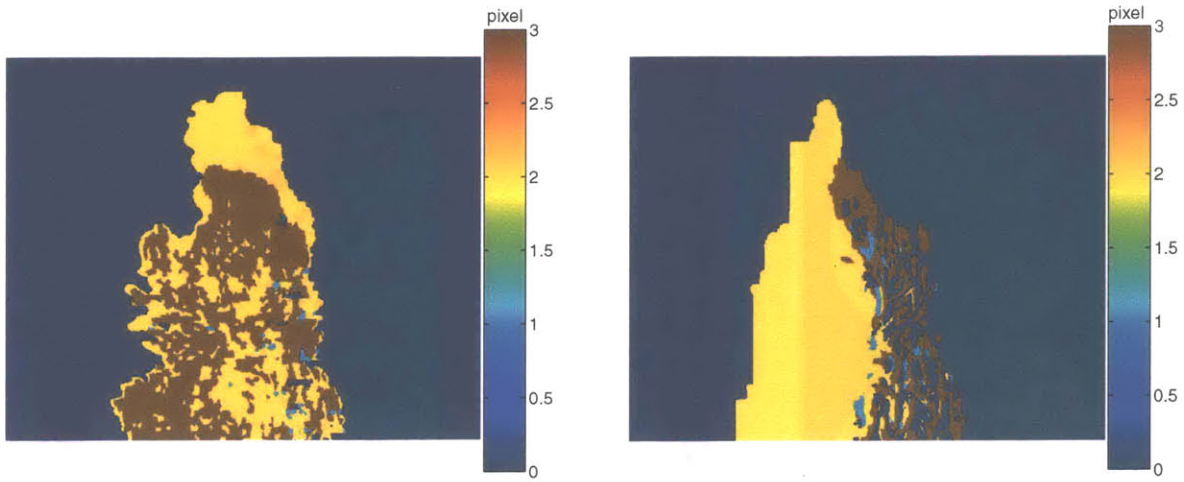

Figure 5-6: Graph cut output from"Reef" image pair. Left: modified energy model with priors. Right: standard energy model.

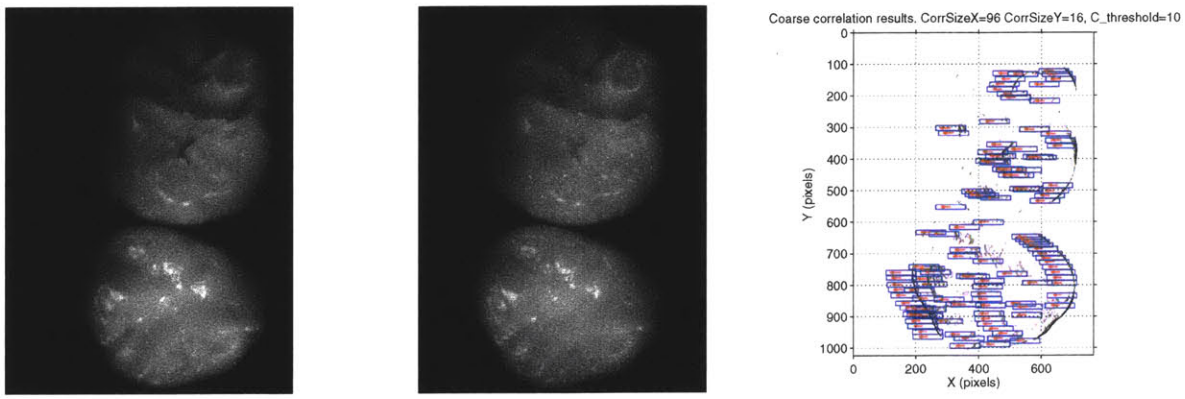

Figure 5-7: Left: left view reference image of "Teeth". Middle: right view. Right: Compressed Feature Correlation outputs.

5.2.1 Accuracy

Table 5.2 compares the results from both standard graph cut and the proposed hybrid algorithm on four image pairs in three categories and the statistics are illustrated in Figure 5-12 and 5-13. Error rates are computed as the number of invalid pixels divided by the total number of pixels in one category. Invalid pixels are defined as those whose output disparities are larger than ± 1 *pixel* of ground truth.

All error rates have decreased in every category except for the "Map" pair. The reason is that compressed feature correlation dose not find enough valid prior labels in the "Map"

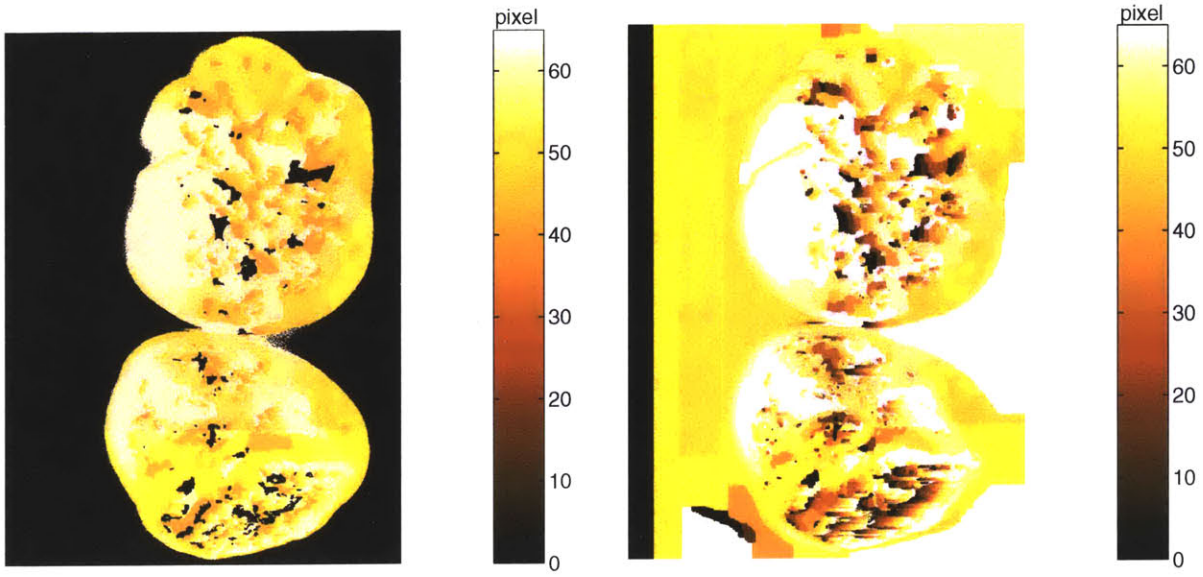

Figure 5-8: Graph cut output from "Teeth" image pair. Left: modified energy model with priors. Right: standard energy model.

$E_{data}(f_p)$	$p \in P - Q$	$p \in Q$		
		$\alpha = f_p$	$\alpha = f_p \pm 1$	else
E_s	$\|I(p) - I'(p + f_s)\|^2$	$\|I(p) - I'(p + \alpha)\|^2$	5	30
E_t	$\|I(p) - I'(p + f_t)\|^2$	$\|I(p) - I'(p + \alpha)\|^2$	0	0

Table 5.1: Data energy definition used in Section 5.2.

image to be statistically significant. In other words, the few prior information contributes little to graph cut. The hybrid approach has a lower accuracy on the "Map" dataset because it only runs one iteration while standard graph cut takes three.

5.2.2 Speed

Computation time using the hybrid approach is usually reduced to less than one third of standard graph cut as shown in Table 5.3 and Figure 5-14 by using only one iteration rather than three. Again, "Map" is an exception because convergence is reached faster than other less densely textured images.

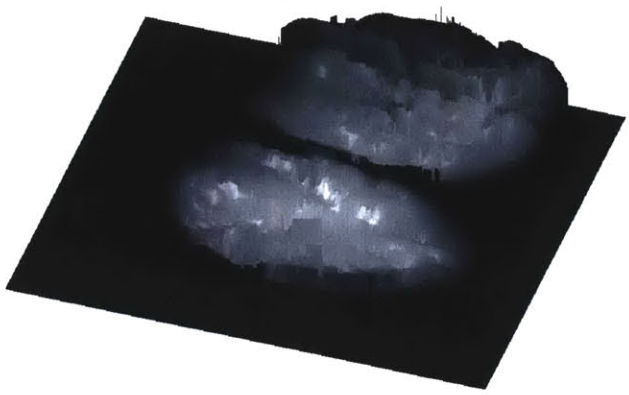

Figure 5-9: Texture mapped rendering of "Teeth" output using modified energy model with priors.

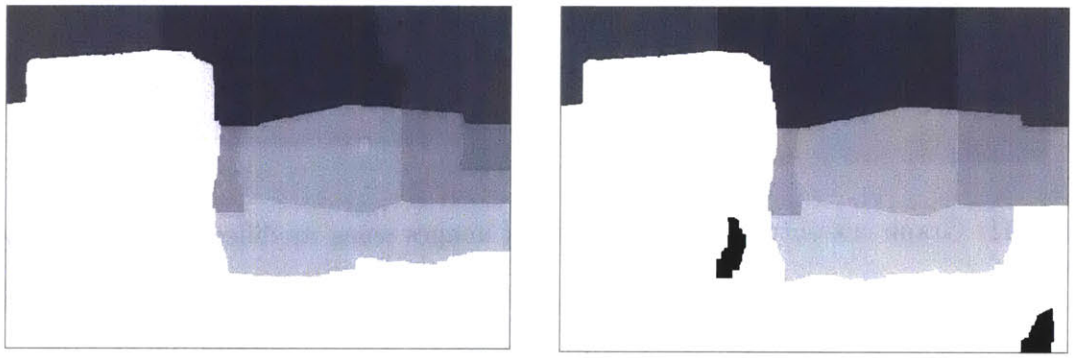

Figure 5-10: Graph cut output for the "Box" image pair. Left: modified energy model with priors. Right: standard energy model.

5.3 Discussions

5.3.1 Label selection

An estimate of disparity range is one of the user inputs to both standard graph cut and the hybrid approach. Minimum disparity is usually set to zero and maximum disparity is Δ. However, Compressed Feature Correlation can serve as a closer disparity range selection process. The argument is that the detected disparities of major features represent most of the labels present in the final graph cut output. This assumption has been proved reasonable in all the datasets studied so far as illustrated by the accuracy improvement in Table 5.2

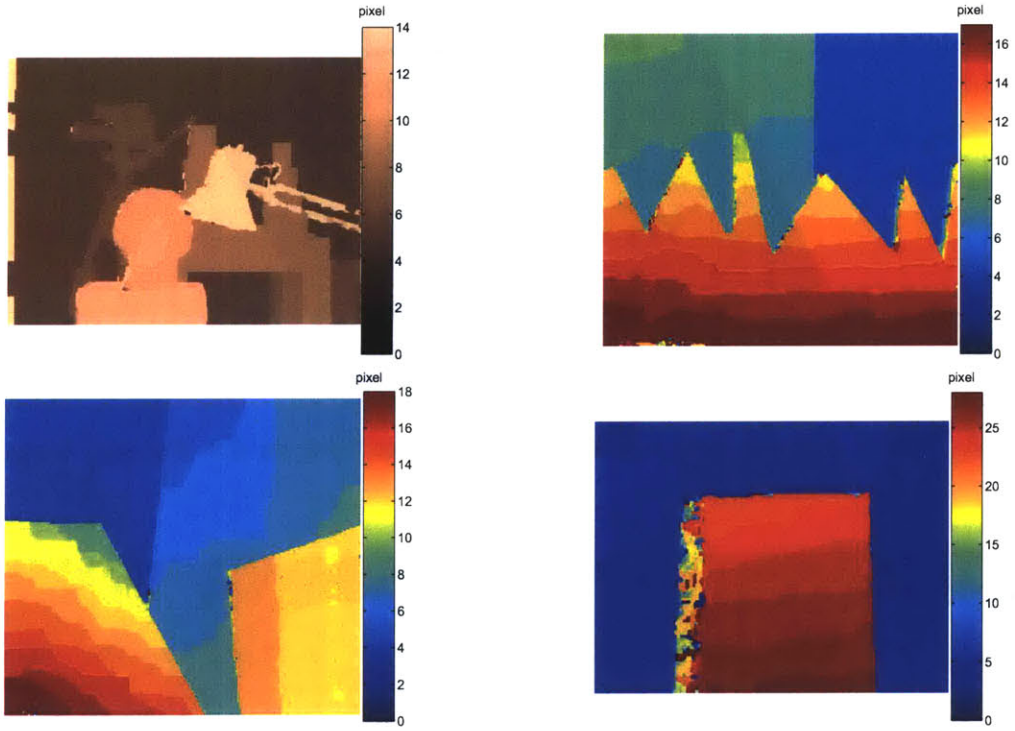

Figure 5-11: Graph cut outputs from benchmark images using modified energy model with priors.

except "Map". A label pre-selection process has been implemented in the hybrid algorithm. The sub-pixel disparity estimations from Compressed Feature Correlation are rounded to the nearest integers towards both positive and minus infinity. The following graph cut only cuts labels that have been validated by compressed correlation. In order to accommodate densely featured datasets with large disparity range such as the "Map", all labels in the $0 - \Delta$ range are cut when the number of detected labels after correlation is less than one

Algorithm	Tsukuba			Sawtooth			Venus			Map	
(%)	all	untex.	disc.	all	untex.	disc.	all	untex.	disc.	all	disc.
Graph cut	1.86	1.00	9.35	0.42	0.14	3.76	1.69	2.30	5.40	0.36	3.91
Hybrid approach	1.53	0.45	8.19	0.30	0.04	2.72	0.57	0.49	4.67	0.49	5.68
Improvement	18%	55%	12%	29%	71%	28%	66%	79%	14%	-36%	-45%

Table 5.2: Error statistics of standard graph cut and graph cut with prior.

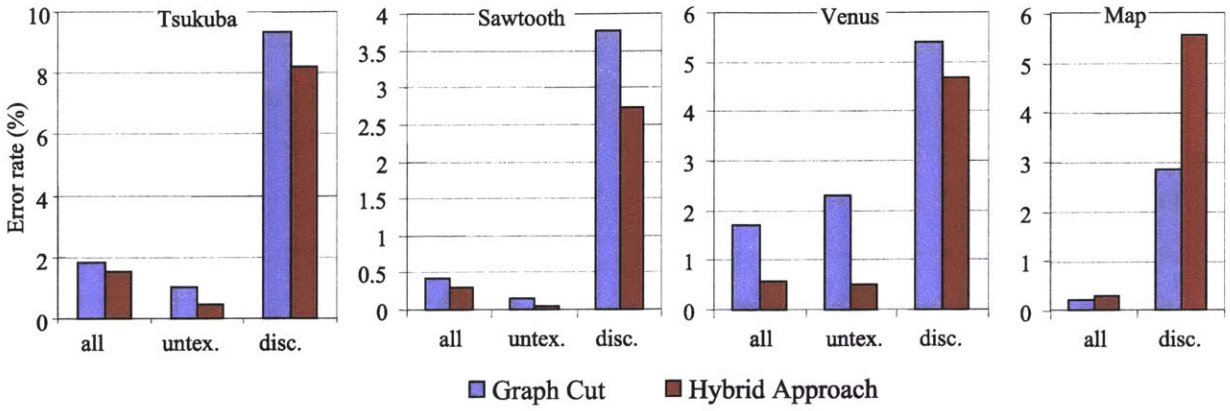

Figure 5-12: Error statistics of standard graph cut and hybrid approach.

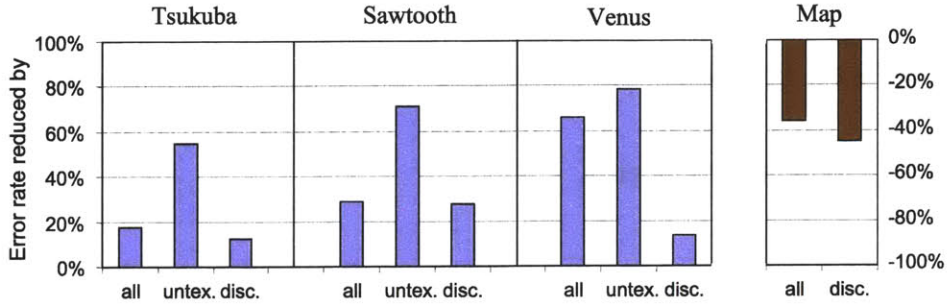

Figure 5-13: Accuracy Improvement.

third of all possible labels.

A smaller label selection affects both speed and accuracy. For example, the disparity range of "Teeth" is 0 to 65 *pixel*, *i.e.*, 66 labels to cut. It takes six minutes for standard graph cut to iterate through 66 labels three times. In comparison, there are only 19 labels present in the compressed correlation outputs in the range from 40 to 62 pixels. Cutting these 19 labels once while setting initial conditions to all zero takes 23 seconds, which is

(second)	Tsukuba	Sawtooth	Venus	Map
Graph cut	7	10.8	15	6
New hybrid approach	1.8	3.2	4.5	3
Improvement	74%	70%	70%	50%

Table 5.3: Processing time of standard graph cut and hybrid approach.

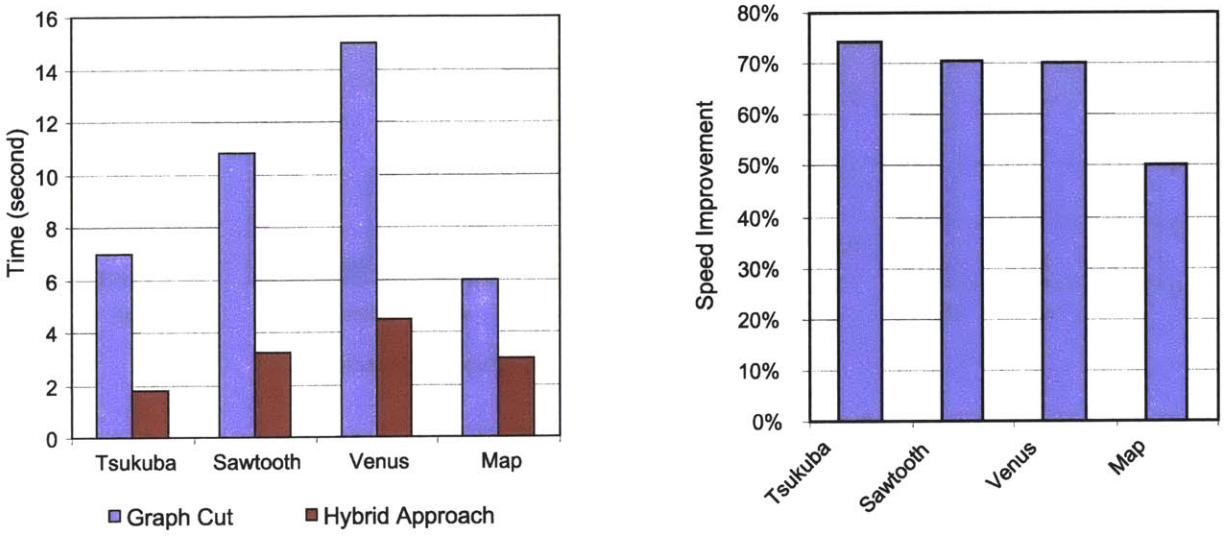

Figure 5-14: Processing time of standard graph cut and hybrid approach.

6.4% the time of standard graph cut.

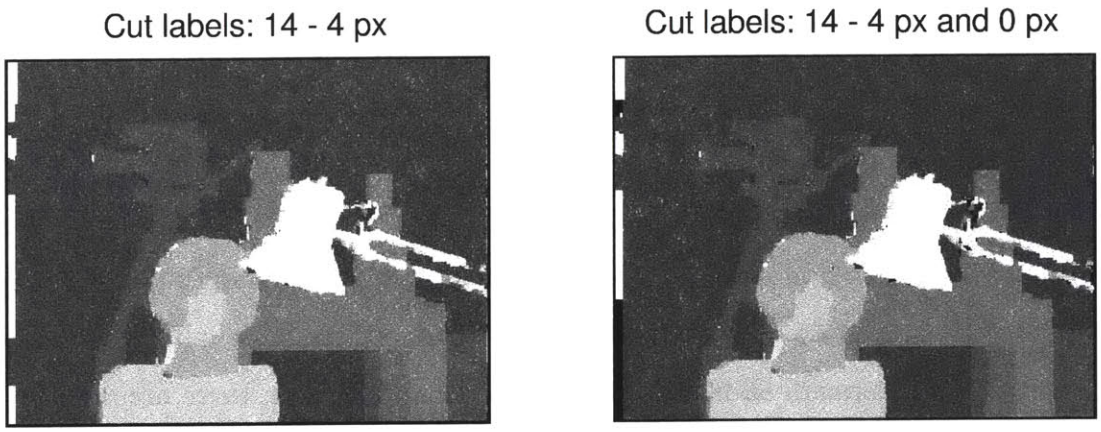

Figure 5-15: Graph cut outputs from "Tsukuba" image pair using modified energy model.

It is important in terms of accuracy to have a close disparity range estimation before graph cut, even using the modified energy model with priors. Figure 5-15 gives one such example. The user specified disparity range is 0-15 pixels. A closer estimation by compressed correlation is 4-14 pixels. The hybrid approach results of two settings are shown: only cutting 11 labels from 4 to 14 pixels or 12 labels with an extra label 0. We can see that overall

that overall accuracy deteriorates after cutting label 0. This extra label erodes into regions without priors, such as the black spots and strips around the lamp and in the background. Global energy is minimized by assigning label 0 to these areas due to data energy noise. The following rule applies to graph cut: the less redundant labels to cut, the less chance of gross errors.

5.3.2 Sensitivity to parameters

The standard energy model depends on four numbers: data energy coefficient n, smoothness penalty K and γ, and static cue threshold $S\_threshold$. The question of how to choose good parameters is not well understood in many energy model based algorithms including graph cut until some recent studies [145]. Common consensus is that data and smoothness energies are balanced when the parameters fall into some specific range. However, if parameters are picked very far away from this range, the results become either over-smoothed or too noisy. Different n calls for different range of smoothness parameters to avoid the above two extremes. Optimum parameter settings may differ significantly for various image types which can probably be explained by varying amount of surface texture density, disparity discontinuities and signal to noise ratio. Using priors generally reduces standard graph cut's sensitivity to parameter settings in its energy model.

In order to evaluate the hybrid algorithm's sensitivity to parameter settings, each parameter is varied in turn while other ones are fixed. For example, $S\_threshold$ varies a lot for different image types depending on local contrast around disparity discontinuities. Figure 5-16 illustrates how error statistics are influenced by $S\_threshold$ with or without priors. Y-axis represents the total error rate in percentage. Standard deviation is 0.7% for standard graph cut, while only 0.1% for the hybrid approach for the "Tsukuba" pair. Standard deviation statistics for all three benchmark images are listed in Table 5.4.

Two new parameters are introduced in the modified energy model as fixed data penalties for priored pixels: W and σ. W is set to $4 - 10$ and $\sigma = 5 - 10$ empirically. If prior information is known to be 100% accurate, the upper limit of σ is INFINITY. By setting

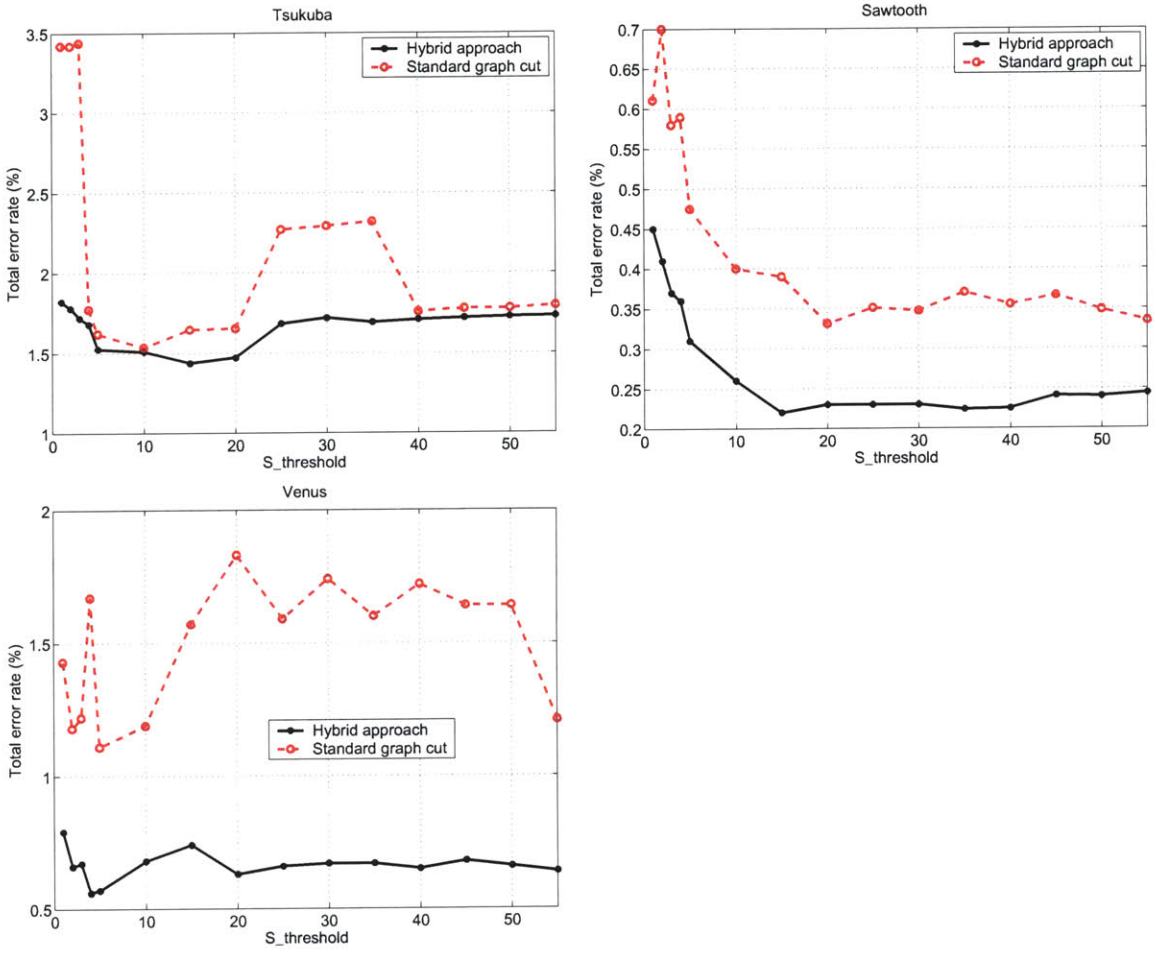

Figure 5-16: Total error rate *vs.* *S_threshold* using both standard and modified energy model.

$\sigma W = INFINITY$, prior information becomes a hard constraint rather than a soft one. However, compressed correlation may generate erroneous disparity estimations around depth discontinuities. A moderate σ gives the few bad priors some freedom to change labels depending on their neighboring pixel labelings. Generally speaking, a setting of $\sigma W \leq 100$ reduces the risk of being trapped with bad priors. The number of discontinuity errors in the compressed correlation outputs is usually so small that manually setting them to the correct values only improves hybrid results marginally.

Compressed Feature Correlation introduces several new parameters to the hybrid ap-

std (%)	Tsukuba	Sawtooth	Venus
Graph cut	0.70	0.12	0.24
New hybrid approach	0.12	0.08	0.06

Table 5.4: Standard deviation statistics of three benchmark images.

proach, such as *C_threshold*, μ and λ of confidence measure, w and h of correlation window size. Varying these parameters within a reasonable range does not affect the final output much. If falling outside the good range, accuracy worsens very fast. The reason is that Compressed Feature Correlation can no longer reliably detect disparities of major features and some important labels are missing when running graph cut.

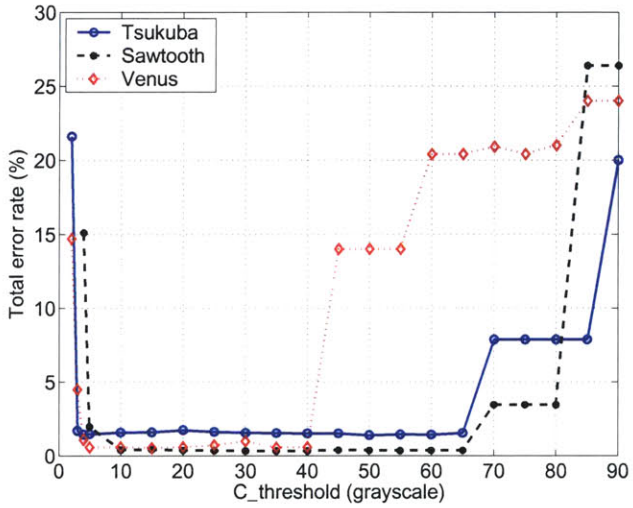

Figure 5-17: Total error rate *vs.* compression threshold using modified energy model.

An important parameter when computing prior disparities for graph cut is the compression ratio *C_threshold*. Figure 5-17 demonstrates that the final output is not sensitive to *C_threshold* in a large range for the three benchmark images. When compression threshold is too low or too high, compressed correlation can no longer generate a reliable disparity range estimation because too few correct correlation vectors can be obtained. An intuitive range selection criterion for *C_threshold* is when the resulting compression ratio is larger than 1%.

5.3.3 Untextured regions

We can see from Figure 5-13 that error rates in untextured regions have a more significant improvement than those in discontinuous regions. This observation clearly demonstrates the advantage of the hybrid approach: correct prior disparities of strong features can be successfully propagated into untextured areas. Priored pixels serve as anchor points when optimizing surrounding regions.

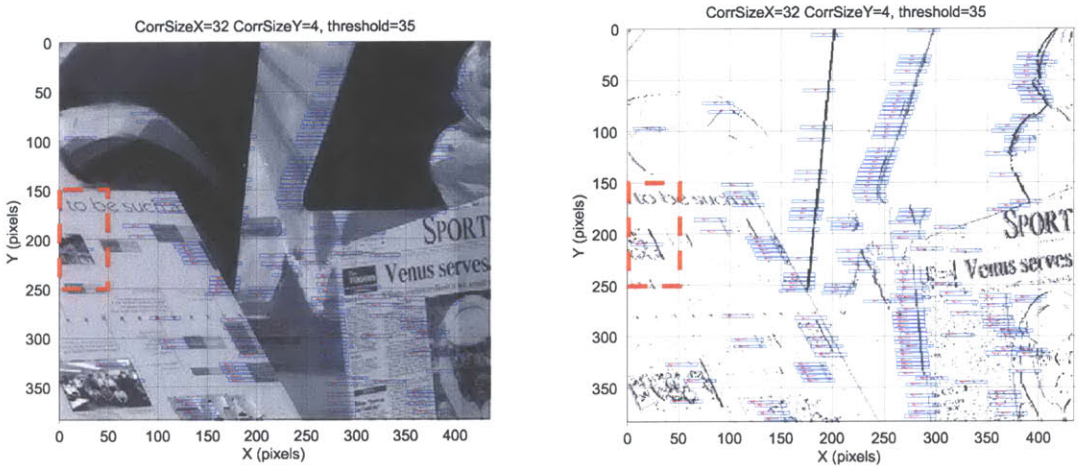

Figure 5-18: Compressed Feature Correlation outputs from "Venus". Left: reference image. Right: compressed view.

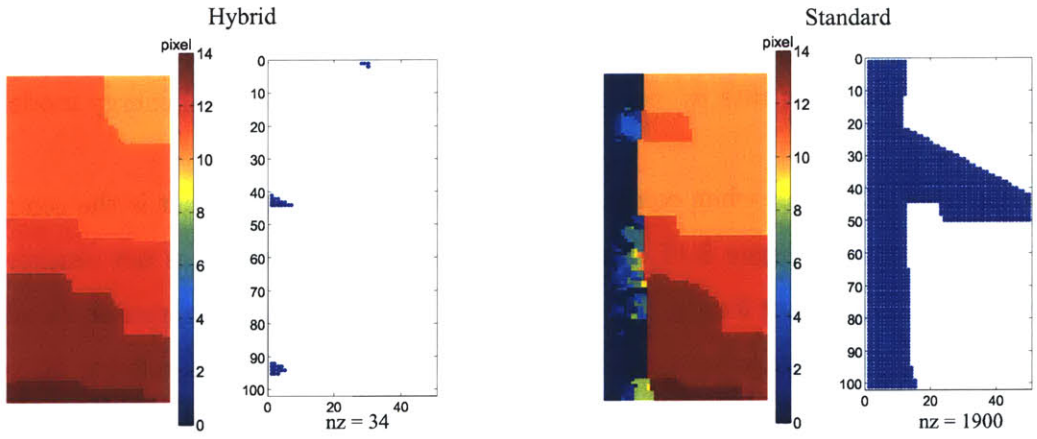

Figure 5-19: Disparity maps and errors of the blowup region in Figure 5-18. Left: hybrid approach. Right: standard graph cut.

Figure 5-18 provides a detailed example of the hybrid approach's performance in untextured regions. The dashed square area in Figure 5-18 is under scrutiny. There are 14 pixels with prior information in this 50×100 *pixel* region. Figure 5-19 shows graph cut results of both the hybrid approach and standard graph cut. Pixels with erroneous disparities are represented by a dot. Some of the errors around the image left boundary in the standard graph output can be attributed to boundary effects and cutting three extra labels from 0 to 2 *pixel* which the hybrid approach is exempt of. However, most errors happen in untextured regions. The hybrid algorithm significantly reduces the number of untextured errors in this example.

5.3.4 Limitations

The hybrid method has certain limitations inherited from its two composing modules, even though their significance has reduced by some degree in the integrated algorithm.

The hybrid approach is not suitable for images which (a) are densely textured and (b) have a large disparity range such as "Map". It works better than the standard graph cut on images that only satisfy one of the conditions, such as "Tsukuba" and "Teeth". Condition (a) and (b) together limit the number of valid windows detected by Compressed Feature Correlation. Too few priored pixels prohibit a realistic disparity range estimation and are not sufficient to stabilize the global optimization process.

Resolution of the hybrid algorithm is limited to integer pixel disparities like standard graph cut. Sub-pixel discretization of disparity values is possible if speed is not a concern. Computational complexity of state of the art graph cut algorithms is still polynomial, not linear. A disparity resolution increase from integer to 0.1 pixel in one direction might result in a computing time $O(10^n)$ times of the original one, where n stands for the order of polynomial. Another solution to achieve sub-pixel labels is by upsampling the input image pair.

5.4 Summary

In conclusion, the effectiveness and efficiency of the proposed hybrid approach have been proved by qualitative and quantitative datasets. Outputs from Compressed Feature Correlation serve as control points to guide the global optimization process. As a result, stability and efficiency of graph cut are improved significantly with reduced computational cost. The hybrid algorithm has three major advantages: improved accuracy by taking advantage of prior information, reduced computational time and improved robustness to parameter settings.

Chapter 6

Conclusions

This dissertation has developed a fast and robust algorithm to solve the dense correspondence problem by merging Sparse Array Correlation from the computational fluids community with graph-based stereo from the computer vision community. In this chapter, we summarize our contributions and point out potential future directions.

6.1 Contributions

This dissertation presents a new method which consists of two independent modules: Compressed Feature Correlation and graph cut with priors.

The first module is called Compressed Feature Correlation, which combines feature-based 3D matching with compressed image correlation. The algorithm uses an image compression scheme that retains pixel values in high intensity gradient areas while eliminating pixels with little correlation information in smooth surface regions. The result is a highly reduced image dataset with lowered computational load. In addition, by utilizing an error correlation function, pixel comparisons are made through single integer calculations eliminating time consuming multiplication and floating point arithmetic. Unlike the traditional fixed window sorting scheme, adaptive correlation window positioning is implemented by dynamically placing strong features at the center of each correlation window. A confidence measure

is developed to validate correlation outputs. The sparse disparity map generated by this extremely fast Compressed Feature Correlation algorithm may either serve as inputs to global methods or suffice to be interpolated into dense disparity map when object boundaries can be clearly detected.

The second module is a modified graph cut algorithm with an improved energy model that accepts prior disparity information by fixing data energy terms. The image pixels with known disparity values help stabilize and speed up global optimization. As a result only one graph cut iteration is necessary instead of the common practice of three and sensitivity to parameters is reduced. Prior information may come from either user input or 3D tracking algorithms.

An efficient hybrid algorithm is implemented based on the above two modules. By coupling a simpler and much less expensive local algorithm, Compressed Feature Correlation, with an expensive global method, graph cut, the computational expense of the hybrid calculation is one third of performing the entire calculation using the more expensive of the two algorithms, while accuracy and robustness are improved.

6.2 Suggestions for future work

By no means all potential avenues of the hybrid approach have been explored. In fact, this dissertation barely opened a new door by bringing together two previously-isolated research communities. Following are some interesting directions worth further pursuing.

Graph cut algorithms have produced strong results in multi-camera scene reconstruction with consideration for occlusion as shown in literature. Introducing priors into these much more sophisticated energy models might greatly speed the algorithms up.

Layered methods overcome the integer pixel limitation of graph cut by interacting between a graph cut module and a surface fitting module. Our hybrid approach might be used to speed up the graph cut module.

It is also interesting to discard the compressed feature correlation step and directly build feature detection into the energy model. Noisy data energy calculation can be verified against

local gradient.

The outputs of compressed feature correlation may also serve as ground control points to one- or two-pass dynamic programming to improve its performance.

So far the essence of Particle Image Velocimetry (PIV) has been introduced to computer vision. The reverse direction might be equally rewarding. PIV algorithms have been troubled with the window averaging effects of local correspondence methods. The hybrid approach presented in this dissertation might be a potential solution.

A final but not least issue that needs to be addressed is one that is inherent to many global optimization algorithms, namely, how to automatically detect image type and select parameters.

Bibliography

[1] Adrian, Ronald J. Particle-imaging techniques for experimental fluid mechanics. *Annual Review of Fluid Mechanics*, 23:261–304, 1991.

[2] Atick, Joseph J., Paul A. Griffin and A. Norman Redlich. Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images. *Neural Computation*, 8(6):1321–1340, 1996.

[3] Baker, Simon, Richard Szeliski and P. Anandan. A layered approach to stereo reconstruction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, pages 434–441, Santa Barbara, CA, 1998.

[4] Barron, J. L., D. J. Fleet, S. S. Beauchemin and T. A. Burkitt. Performance of optical flow techniques. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'92)*, Champaign, IL, USA, 1992.

[5] Bartoli, A., R. Hartley and F. Kahl. Motion from 3D line correspondences: Linear and non-linear solutions. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Madison, Wisconsin, 2003.

[6] Birchfield, S. and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, 1998.

[7] Birchfield, S. and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, 1999.

[8] Birchfield, S. T. *Depth and Motion Discontinuities.* PhD thesis, Stanford University, Department of Electrical Engineering, 1999.

[9] Birchfield, Stanley T. Depth Discontinuities by Pixel-to-Pixel Stereo. http://vision.stanford.edu/ birch/p2p/.

[10] Blais, F. and Marc Rioux. Biris: A simple 3-D sensor. *SPIE Proceedings*, 728:235–242, 1986.

[11] Blake, Andrew and Andrew Zisserman. *Visual Reconstruction.* MIT Press, Cambridge, MA, 1987.

[12] Boykov, Y. and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124 – 1137, 2004.

[13] Boykov, Y., O. Veksler and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[14] Boykov, Yuri, Olga Veksler and Ramin Zabih. Markov random fields with efficient approximations. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR'98)*, Santa Barbara, CA, 1998.

[15] Brown, M. Z., D. Burschka and G. D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):993– 1008, 2003.

[16] Brunelli, R. and T. Poggio. Face recognition: features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[17] Carnegie Mellon University. The Robotics Institute. Vision and Autonomous Systems Center's Image Database. http://vasc.ri.cmu.edu/idb/html/stereo/arch/.

[18] Chalupa, Leo M. and John S. Werner. *The visual neurosciences.* MIT Press, Cambridge, MA, 2004.

[19] Chen, Frank, Gordon M. Brown and Mumin Song. Overview of three-dimensional shape measurement using optical methods. *Optical Engineering*, 39(1):10–22, 2000.

[20] Chien, Sung-Il and Si-Hun Sung. Adaptive window method with sizing vectors for reliable correlation-based target tracking. *Pattern Recognition Letters*, 33(2):237–249, 2000.

[21] Cook, William J., William H. Cunningham, William R. Pulleyblank and Alexander Schrijver. *Combinatorial Optimization.* Wiley-Interscience, 1997.

[22] Coppin, Ben. *Artificial intelligence illuminated.* Jones and Bartlett Publishers, Boston, MA, 2004.

[23] Cormen, T. H., C. E. Leiserson and R. L. Rivest. *Introduction to Algorithms.* MIT Press and McGraw-Hill, 1992.

[24] Davis, C. Q. and D. M. Freeman. Statistics of subpixel registration algorithms based on spatiotemporal gradients or block matching. *Optical Engineering*, 37(4):1290–1298, 1998.

[25] Deriche, R. and O. D. Faugeras. Tracking line segments. In *European Conference on Computer Vision (ECCV'90)*, Antibes, France, 1990.

[26] Desouza, G. N. and A. C. Kak. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237 267, 2002.

[27] Drummond, T. and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):932–946, 2002.

[28] Elder, J. H. Are edges incomplete? *International Journal of Computer Vision*, 34(2-3):97–122, 1999.

[29] Enns, James T. *The thinking eye, the seeing brain : explorations in visual cognition.* W. W. Norton, New York, 2004.

[30] Ens, John and Peter Lawrence. A matrix based method for determining depth from focus. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '91)*, Maui, HI, 1991.

[31] Fua, P. Reconstructing complex surfaces from multiple stereo views. In *Fifth International Conference on Computer Vision (ICCV'95)*, Cambridge, MA, 1995.

[32] Galvin, B., B. McCane, K. Novins, D. Mason and S. Mills. Recovering motion fields: An evaluation of eight optical flow algorithms. In *British Machine Vision Converence (BMVC'98)*, Southampton, England, 1998.

[33] Gao, Y. and M. K. H. Leung. Face recognition using line edge map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):764–779, 2002.

[34] Geiger, D. and A. Yuille. A common framework for image segmentation. *International Journal of Computer Vision*, 6(3):227–243, 1991.

[35] Geiger, Davi, Bruce Ladendorf and Alan Yuille. Occlusions and binocular stereo. *International Journal of Computer Vision*, 14(3):211–226, 1995.

[36] Gong, M. and Y.-H. Yang. Genetic-based stereo algorithm and disparity map evaluation. *International Journal of Computer Vision*, 47(1-3):63–77, 2002.

[37] Goodman, Joseph W. *Introduction To Fourier Optics.* McGraw-Hill Science/Engineering/Math, New York, 2 edition, 1996.

[38] Gregory, R. L. *Eye and brain: the psychology of seeing.* Princeton University Press, Princeton, NJ, 1997.

[39] Greig, D., B. Porteous and A. Scheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989.

[40] Grimson, W. E. L. Computational experiments with a feature based stereo algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:17–34, 1985.

[41] Gunn, Steve R. and Mark S. Nixon. A robust snake implementation; a dual active contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):63 68, 1997.

[42] Habibi, B. and F. Maslar. Single camera 3DTM (SC3DTM): a novel technology for guidance of industrial robots in three-dimensional space. *Braintech*, pages 1–19, 2002.

[43] Han, M. and T. Kanade. Multiple motion scene reconstruction with uncalibrated cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):884–894, 2003.

[44] Harding, Cressida M. *How far away is it? Depth estimation by a moving camera.* PhD thesis, University of Canterbury, Department of Electrical and Electronic Engineering, 2001.

[45] Harding, Cressida M., Andrew S. L. Bainbridge-Smith, James T. G. Preddy and Richard G. Lane. Accurate estimate of optical flow by modelling tilted facets. In *Fifth International Conference on Automation, Robotics, Control and Vision*, Singapore, 1998.

[46] Hart, D. P. The elimination of correlation errors in PIV processing. In *9th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, Lisbon, Portugal, 1998.

[47] Hart, D. P. High-speed PIV analysis using compressed image correlation. *Journal of Fluids Engineering-Transactions of the ASME*, 120(3):463–470, 1998.

[48] Hart, D. P. PIV error correction. *Experiments in Fluids*, 29(1):13–22, 2000.

[49] Hart, D. P. Super-resolution PIV by recursive local-correlation. *Journal of Visualization*, 3(2):187–194, 2000.

[50] Hart, D. P. PIV processing using multidimensional correlation. In *The 10th International Symposium on Flow Visualization*, Kyoto, Japan, 2002.

[51] Hirschmuller, Heiko, Peter R. Innocent and Jon Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1/2/3):229–246, 2002.

[52] Hoff, W. and N. Ahuja. Surfaces from stereo: integrating feature matching, disparity estimation, and contour detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):121 136, 1989.

[53] Holst, Gerald C. *CCD arrays, cameras, and displays.* Winter Park, FL : JCD Publishing, Bellingham, WA, 1998.

[54] Horn, B. K. P. and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[55] Horn, B. K. P. and B. G. Schunck. Determining optical flow - a retrospective. *Artificial Intelligence*, 59(1-2):81–87, 1993.

[56] Horn, B.K.P. *Robot Vision.* MIT Press, Cambridge, MA, 1986.

[57] Hoznek, A., S. Zaki, D. Samadi, L. Salomon, A. Lobontiu, P. Lang and C.-C. Abbou. Robotic assisted kidney transplantation: an initial experience. *Journal of Urology*, 167(4):1604 6, 2002.

[58] Hsu, Feng-hsiung, Murray S. Campbell and A. Joseph Hoane, Jr. Deep blue system overview. In *Proceedings of the 9th international conference on Supercomputing*, Barcelona, Spain, 1995.

[59] Hung, Y. Y., L. Lin, H. M. Shang and B. G. Park. Practical three-dimensional computer vision techniques for full-field surface measurement. *Optical Engineering*, 39(1):143–149, 2000.

[60] Ishikawa, H. and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *European Conference on Computer Vision (ECCV'98)*, Freiburg, Germany, 1998.

[61] Izquierdo, M. E. Disparity segmentation analysis: Matching with an adaptive window and depth-driven segmentation. *IEEE Trans. Circuits and Systems for Video Technology*, 9(4):589–607, 1999.

[62] Jeong, Hong and Yuns Oh. Parallel trellis based stereo matching using constraints. In *IEEE International Workshop on Biologically Motivated Computer Vision (BMCV'00)*, Seoul, Korea, 2000.

[63] Jin, Lianhua, Yutaka Kodera, Toru Yoshizawa and Yukitoshi Otani. Shadow moiré profilometry using the phase-shifting method. *Optical Engineering*, 39(8):2119–2123, 2000.

[64] Kanade, T. and M. Okutomi. A stereo matching algorithm with an adaptive window: theory and experiment. *IEEE Trans. Pattern Analyze and Machine Intelligence*, 16(9):920–932, 1994.

[65] Kanade, Takeo. Recovery of the three dimensional shape of an object from a single view. *Artificial Intelligence*, 17(1), 1981.

[66] Kaufmann, Guillermo H. and Pierre Jacquot. Phase shifting of whole field speckle photography fringes. *Applied Optics*, 29(25):3570–3571, 1990.

[67] Kim, C., K. M. Lee, B. T. Choi, and S. U. Lee. A dense stereo matching using two-pass dynamic programming with generalized ground control points. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, 2005.

[68] Kirkpatrick, S., C. Gellat and M. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[69] Kolmogorov, V. *Graph Based Algorithms for Scene Reconstruction from Two or More Views.* PhD thesis, Cornell University, Department of Computer Science, 2004.

[70] Kolmogorov, V. and R. Zabih. Computing visual correspondence with occlusions via graph cuts. In *IEEE International Conference on Computer Vision (ICCV'01)*, Vancouver, Canada, 2001.

[71] Kolmogorov, V. and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conference on Computer Vision (ECCV'02)*, Copenhagen, Spain, 2002.

[72] Kolmogorov, V. and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[73] Kolmogorov, Vladimir. Software. http://www.cs.cornell.edu/People/vnk/software.html.

[74] Leclerc Y. and A. Bobick. The direct computation of height from shading. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'91)*, 1991.

[75] Lee, D. and I. Kweon. A novel stereo camera system by a biprism. *IEEE Transactions on Robotics and Automation*, 16(5):528 – 541, 2000.

[76] Leymarie, F. and M. D. Levine. Tracking deformable objects in the plane using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):617 – 634, 1993.

[77] Lilley, Francis, Michael J. Lalor and David R. Burton. Robust fringe analysis system for human body shape measurement. *Optical Engineering*, 39(1):187–195, 2000.

[78] Lim, Jae S. *Two-Dimensinal Signal and Image Processing.* Prentice Hall, Upper Saddle River, NJ, 1990.

[79] Lin, M. H. and C. Tomasi. Surfaces with occlusions from layered stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Madison, Wisconsin, 2003.

[80] Lin, Michael H. *Surfaces with Occlusions from Layered Stereo.* PhD thesis, Stanford University, Department of Computer Science, 2002.

[81] Liu, Weiyi, Zhaoqi Wang, Guoguang Mu and Zhiliang Fang. Color-coded projection grating method for shape measurement with a single exposure. *Applied Optics*, 39(20):3504–3508, 2000.

[82] Liu, Yen-Fu. *A Unified Approach to Image Focus and Defocus Analysis.* PhD thesis, SUNY at Stony Brook, Department of Electrical and Computer Engineering, 1998.

[83] Lowe, David G. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[84] Lu, Minfu, Xiaoyan He and Sheng Liu. Powerful frequency domain algorithm for frequency identification for projected grating phase analysis and its applications. *Optical Engineering*, 39(1):137–142, 2000.

[85] Luo, P. F. and S. S. Liou. Measurement of curved surface by stereo vision and error analysis. *Optics and Lasers in Engineering*, 30:471–486, 1998.

[86] Maciel, J. and J. P. Costeira. A global solution to sparse correspondence problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):187–199, 2003.

[87] Mahamud, S., L. R. Williams, K. K. Thornber and K. Xu. Segmentation of multiple salient closed contours from real images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4):433–444, 2003.

[88] Mansouri, A.-R. Region tracking via level set pdes without motion computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):947–961, 2002.

[89] Marr, David and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194(4262):283–287, 1976.

[90] Molavi, Diana Weedman. Eye and retina. http://thalamus.wustl.edu/course/eyeret.html.

[91] Moravec, Hans. When will computer hardware match the human brain? *Journal of Transhumanism*, 1, 1998.

[92] Mühlmann, K., D. Maier, J. Hesser and R. Männer. Calculating dense disparity maps from color stereo images, an efficient implementation. *International Journal of Computer Vision*, 47(1-3):79–88, 2002.

[93] Nagle, M. G. and M. V. Srinivasan. Structure from motion: determining the range and orientation of surfaces by image interpolation. *Journal of the Optical Society of America A*, 13(1):25 34, 1996.

[94] Narayanan, P. J., P. W. Rander and T. Kanade. Constructing virtual worlds using dense stereo. In *6th International Conference on Computer Vision (ICCV'98)*, Bombay, India, 1998.

[95] Nayar, Shree K., Masahiro Watanabe and Minori Noguchi. Real-time focus range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1186–1996, 1996.

[96] Nishino, Ko and Shree K. Nayar. The world in an eye. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*, Washington, D.C., USA, 2004.

[97] Oh, B. M., M. Chen, J. Dorsey and F. Durand. Image-based modeling and photo editing. In *SIGGRAPH*, 2001.

[98] Ohta, Yuichi and Takeo Kanade. Stereo by intra-and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.

[99] Okutomi, M. and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353 – 363, 1993.

[100] Okutomi, M., Y. Katayama and S. Oka. A simple stereo algorithm to recover precise object boundaries and smooth surfaces. *International Journal of Computer Vision*, 47(1-3):261 – 273, 2002.

[101] Olsen, S. I. Stereo correspondence by surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):309–315, 1990.

[102] Otten, R. H. J. M. and L. P. P. P. van Ginneken. *The annealing algorithm*. Kluwer Academic Publishers, Boston, MA, 1989.

[103] Paris, S., F. Sillion and L. Quan. A surface reconstruction method using global graph cut optimization. In *Asian Conference of Computer Vision*, Jeju Island, Korea, 2004.

[104] Proesmans, Marc and Luc Van Gool. One-shot 3d-shape and texture acquisition of facial data. In *First International Conference on Audio- and Video-Based Biometric Person Authentication*, Crans-Montana, Switzerland, 1997.

[105] Rohály, J. and D. P. Hart. High resolution, ultra fast 3-D imaging. In *SPIE Three-Dimensional Image Capture and Applications III*, pages 2–10, San Jose, CA, 2000.

[106] Rohály, J., J. Lammerding and D. P. Hart. Monocular 3-d magnetic bead microrheometry. In *11th International Symposium on Applications of Laser Techniques to Fluid Mechanics*, Lisbon, Portugal, 2002.

[107] Roth, G., D. P. Hart and J. Katz. Feasibility of using the L64720 video motion estimation processor (MEP) to increase efficiency of velocity map generation for particle image velocimetry (PIV). In *ASME/JSME Fluids Engineering and Laser Anemometry Conference*, Hilton Head, South Carolina, 1995.

[108] Roy, S. and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *International Conference on Computer Vision (ICCV'98)*, Bombay, India, 1998.

[109] Ryan, M. J., R. K. Erickson, D. N. Levin, C. A. Pelizzari, R. L. Macdonald and G. J. Dohrmann. Frameless stereotaxy with real-time tracking of patient head movement and retrospective patient-image registration. *Journal of Neurosurgery*, 85(2):287–292, 1996.

[110] Sansoni, G., L. Biancardi, U. Minoni and F. Docchio. A novel, adaptive system for 3-D optical profilometry using a liquid crystal light projector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(4):558–566, 1994.

[111] Sansoni, G., S. Corini, S. Lazzari, R. Rodella and F. Docchio. Three-dimensional imaging based on Gray-code light projection: characterization of the measuring algorithm and development of a measuring system for industrial applications. *Applied Optics*, 36(19):4463–4472, 1997.

[112] Scarano, F. Iterative image deformation methods in PIV. *Measurement Science and Technology*, 13(1):1–19, 2002.

[113] Scharstein, D. and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[114] Scharstein, Daniel and Richard Szeliski. Middlebury College Stereo Vision Research Page. http://cat.middlebury.edu/stereo/.

[115] Scharstein, Daniel and Richard Szeliski. Stereo matching with nonlinear diffusion. *International Journal of Computer Vision*, 28(2):155–174, 1998.

[116] Sharma, J., V. Dragoi, J. B. Tenenbaum, E. K. Miller and M. Sur. V1 neurons signal acquisition of an internal representation of stimulus location. *Science Magazine*, 300:1758–1763, 2003.

[117] Shi, J. and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, WA, 1994.

[118] Sinclair, Sandra. *How Animals See: Other Visions of Our World.* Facts on File Publications, New York, 1985.

[119] Smith, Warren J. *Modern Optical Engineering.* McGraw-Hill Professional, New York, 3 edition, 2000.

[120] Strand, T. C. Optical three-dimensional sensing for machine vision. *Optical Engineering*, 24(1):33–40, 1985.

[121] Subramanian, Anbumani, Lakshmi R. Iyer, A. Lynn Abbott and Amy E. Bell. Segmentation and range sensing using a moving-aperture lens. *Machine Vision and Applications*, 15(1):46–53, 2003.

[122] Sun, C. Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques. *International Journal of Computer Vision*, 47(1-3):99–117, 2002.

[123] Szeliski, Richard and Polina Golland. Stereo matching with transparency and matting. *International Journal of Computer Vision*, 32(1):45–61, 1999.

[124] Takeda, Mitsuo and Hirokazu Yamamoto. Fourier-transform speckle profilometry: three-dimensional shape measurements of diffuse objects with large height steps and/or spatially isolated surfaces. *Applied Optics*, 33(34):7829–7837, 1994.

[125] Takeda, Mitsuo, Quan Gu, Masaya Kinoshita, Hideaki Takai and Yosuke Takahashi. Frequency-multiplex fourier-transform profilometry: a single-shot three-dimensional shape measurement of objects with large height discontinuities and or surface isolations. *Applied Optics*, 36(22):5347–5354, 1997.

[126] Takeda, Mitsuo, Takahiro Aoki, Yoko Miyamoto, Hideyuki Tanaka, Ruowei Gu and Zhibo Zhang. Absolute three-dimensional shape measurements using coaxial and coim-

age plane optical systems and fourier fringe analysis for focus detection. *Optical Engineering*, 39(1):61–68, 2000.

[127] Tan, Sheng. Particle displacement measurement using optical diffraction. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science; Department of Mechanical Engineering, 2002.

[128] Tan, Sheng. Three-dimensional imaging technique using optical diffraction. *SPIE Proceedings*, 4567:21–28, 2002.

[129] Tan, Sheng S. and D. P. Hart. A fast and robust feature-based 3D algorithm using compressed image correlation. *Pattern Recognition Letters*, 26(11):1620 1631, 2005.

[130] Tang, Li, Chengke Wu and Zezhi Chen. Image dense matching based on region growth with adaptive window. *Pattern Recognition Letters*, 23(10):1169–1178, 2002.

[131] Tang, Shouhong and Yau Y. Hung. Fast profilometer for the automatic measurement of 3-D object shapes. *Applied Optics*, 29(20):3012–3018, 1990.

[132] Tappen, M. F. and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *International Conference on Computer Vision (ICCV'03)*, Nice, France, 2003.

[133] The Editors of Scientific American. *The scientific American book of the Brain.* Scientific American, New York, 1999.

[134] Timoner, S. J. and D. M. Freeman. Multi-image gradient-based algorithms for motion estimation. *Optical Engineering*, 40(9):2003 2016, 2001.

[135] Tommasini, T., A. Fusiello, E. Trucco and V. Roberto. Making good features track better. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'98)*, Santa Barbara, CA, 1998.

[136] Torroba, Roberto D., Luiz Carlos S. Nunes and A. A. Tagliaferri. Object positioning by a digital moiré focusing technique. *Optical Engineering*, 38(8):1409–1412, 1999.

[137] van Laarhoven, P. J. M. and E. H. L. Aarts. *Simulated Annealing: Theory and Application.* Dordrect: Reidel, Amsterdam, Holland, 1987.

[138] Veksler, O. Stereo matching by compact windows via minimum ratio cycle. In *International Conference on Computer Vision (ICCV'01)*, Vancouver, Canada, 2001.

[139] Villa, J., M. Servin and L. Castillo. Profilometry for the measurement of 3-D object shapes based on regularized filters. *Optics Communications*, 161:13–18, 1999.

[140] Westerweel, J. Theoretical analysis of the measurement precision in particle image velocimetry. *Experiments in Fluids*, 29(7):3–12, 2000.

[141] Willert, C. E. and M. Gharib. Three-dimensional particle imaging with a single camera. *Experiments in Fluids*, 12(6):353–358, 1992.

[142] Xiao, J. and M. Shah. Motion layer extraction in the presence of occlusion using graph cut. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'04)*, Washington, D.C., 2004.

[143] Yu, Albert. The future of microprocessors. *IEEE Micro*, 16(6):46–53, 1996.

[144] Zhang, Hong, Fang Wu, Michael J. Lalor and David R. Burton. Spatiotemporal phase unwrapping and its application in fringe projection fiber optic phase-shifting profilometry. *Optical Engineering*, 39(7):1958–1964, 2000.

[145] Zhang, Li and Steven M. Seitz. Parameter estimation for MRF stereo. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, 2005.

[146] Zitnick, C. L. and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):675 – 684, 2000.