A COMPUTER STEREO VISION SYSTEM – USING HORIZONTAL INTENSITY LINE SEGMENTS BOUNDED BY EDGES

By

Chor-tung YAU

A Dissertation

submitted in partial fulfillment of the requirements

for the Degree of Master of Philosophy

Division of Computer Science

The Chinese University of Hong Hong

1996

Acknowledgement

I would like to express my gratitude to my project supervisor, Dr. Kin-hong Wong for his guidance throughout the project. His help is not only technical but also spiritual. I also want to thank the people in the Department of Computer Science and Engineering, The Chinese University of Hong Kong for providing the learning and research atmosphere. I enjoy the chances of discussions with and learning from them.

Abstract

In quest of equipping computer vision systems with the ability of measuring depth information of the 3-D world, binocular stereo vision has long been a topic of research. There are many possibilities of applications such as cartography, object recognition, automation in assembly lines and autonomous vehicles.

A new stereo matching method using horizontal intensity line segment bounded by edges as matching primitive is studied. The validity and advantages of the use of such line segments as matching primitives in stereo vision systems is argued for. Moreover, the use of such segments as matching primitives give rise to a number of favourable features such as reduction of search space (speed-up of matching), detection and disambiguation of inclined surfaces and partial occlusion. The detection of inclined surfaces and partial occlusion is performed simultaneously as part of the matching process, rather than a post-processing step.

The new stereo matching method is studied and implemented. It is tested on a number of synthetic images as well as real images.

Contents

Chapter 1

Introduction

1.1 Objectives

This thesis tackles a well-known problem in the field of computer vision – the stereo vision problem. The stereo vision problem is the reconstruction of 3-D coordinates of points in a scene from multiple images (usually two) obtained by cameras of known relative positions and orientations. It is a practical problem with many possibilities of applications such as cartography, object recognition, automation in assembly lines and autonomous vehicles. The idea of using multiple 2-D images to reconstruct the 3-D coordinates of points in a scene comes from the knowledge of the human visual system.

The first two chapters give an introduction to the problem and some related previous work. The later chapters present a novel stereo algorithm in details and its experimental results.

1.2 Factors of Depth Perception in Human Visual System

Human, as well as many other creatures, are able to perceive the world as a 3-D world in spite of the fact that the images on the retinae are 2-D images. Then how is the depth information encoded in the 2-D images? Researchers in the field of psychology have proposed different answers to this question. The depth information can be determined from different factors. Among the factors, there are oculomotor cues, pictorial cues, movement-produced cues and binocular disparity.

1.2.1 Oculomotor Cues

Oculomotor cues are the information provided by our eye muscles. Two sets of muscles provide the information. They are the eye muscles in our head that control eye movement and the circular muscles that surround our eye lens. When we look at close objects, our eyes converge. This convergence of the eyes cause the eyes to look inwards. At the same time, the muscles that surround our lens contract to change the shape of the lens to a more convex shape so that we can focus on close objects. This is called accommodation of the eye lens. On the other case, when we look at distant objects, our eyes diverse and the muscles of the lens relax, causing the lens to change into its natural (less convex) shape and focus on the distant objects. The convergence and accommodation are cues to depth as the states of the muscles are correlated with the distance between the objects and the eyes.

1.2.2 Pictorial Cues

Pictorial cues are cues that can be depicted in a still picture. They involve high level reasoning processes and understanding of the physical world. They provide qualitative depth information instead of accurate quantitative depth measures.

1. Overlapping

   When object A covers part of object B, object A should be in front of object B. Overlapping indicates relative depth.

2. Size in the Field of View

   A distant object takes up less of our field of view than a close object. When two objects of the same shape are presented, the one with larger size appears to be closer.

3. Height in the Field of View

   Objects below the horizon that appear higher in the field of view are perceived to be further away, while objects appear lower in the field of view are perceived closer. On the other hand, objects above the horizon that appear lower in the field of view are seen as being further away.

4. Atmospheric Perspective

   The atmosphere makes far objects look less sharp and also makes them look blue in color. This is due to the light refraction occurring in the dust particles in the air and in the air itself.

5. Linear Perspective

   Lines that are parallel in the scenes appear to be converging to a same point in the distant. This has long been used by artists in their paintings to present scenes with depth.

1.2.3 Movement-Produced Cues

When we are moving, the images of the objects in our retinae change their positions, and we perceive the objects to be moving. This is because there are relative movement between the observer and the objects. These changes provide cues to depth information.

1. Motion Parallax

   When we move, such as when we look out the side window of a moving car, nearby objects appear to be moving fast in a direction opposite to that of our movement, while distant objects appear to be moving relatively slowly. This difference in speed of movement between nearby and distant objects is called motion parallax. This is an important source of depth information (fig. 1.1).

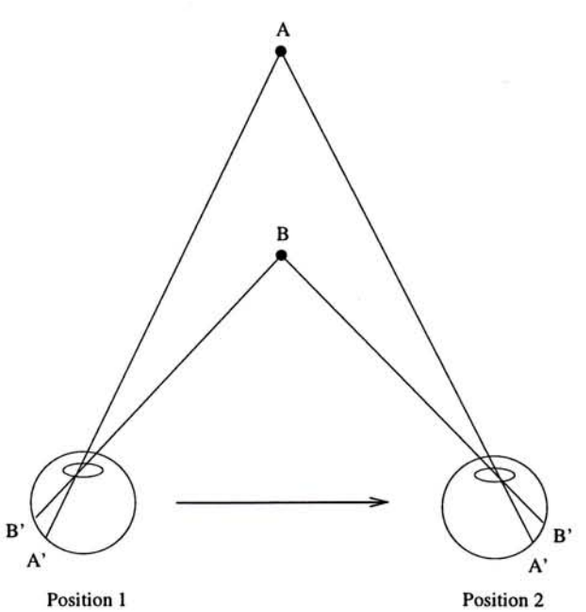


Figure 1.1: Showing how the images of two objects (A and B) change their positions on the retina due to movement of the eye from left to right. Note that the image of the near object, B, moves farther on the retina than that of the far object, A.

2. Deletion and Accretion

   Deletion and accretion occurs when an observer moves in a direction not perpendicular to two surfaces that are at different depth. When the observer moves, either deletion or accretion of the further object by the closer object occurs (see fig. 1.2). The object being deleted or accreted is perceived further away.

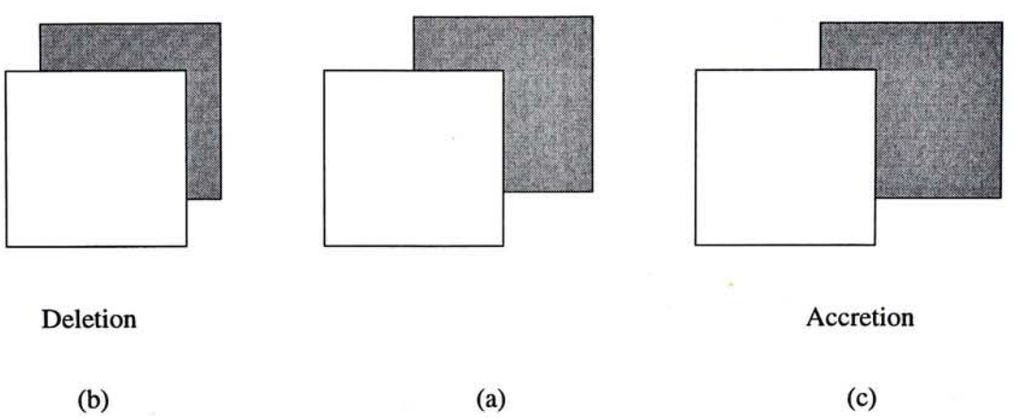


Figure 1.2: Showing deletion and accretion. The white surface is closer to the observer than the grey one. When an observer sees the two surfaces as in (a) and then moves to the left, deletion of the rear object by the near one is observed, as in (b). When the observer starts at (a) and move to the right, accretion occurs, as in (c).

1.2.4 Binocular Disparity

Human are equipped with two eyes. The two eyes see the world from slightly different positions and the images on the two retinae are slightly different. Our brain can compute the depth information from the difference. The fact that our two eyes see different views of the world was used by Charles Wheatstone (1802-1875) to create the stereoscope, a device which projects two slightly different pictures into each eye to produce a convincing illusion of a scene with depth. In 1960, Bela Julesz created the illusion of depth using a stereoscope with random

dot patterns, which is known as the random dot stereogram (fig. 1.3). The random dot stereogram demonstrates that binocular disparity is an important cue to depth and that it alone can cause depth perception. The random dot stereoscope contains no depth information other than disparity. Fig. 1.4 shows that disparity is correlated with depth perception. Details of the mechanism of depth from binocular disparity will be discussed in section 1.4.

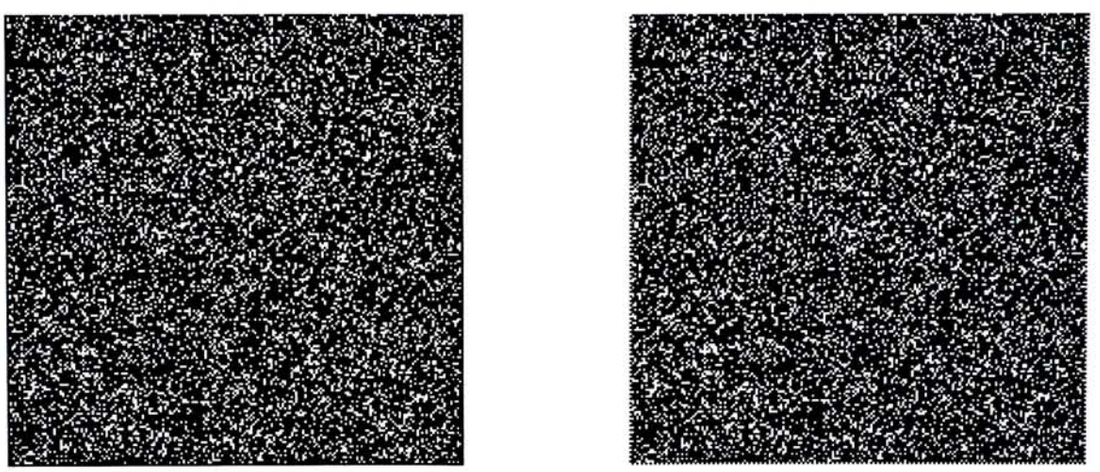


Figure 1.3: Random dot stereogram.

1.3 What Cues to Use in Computer Vision?

Different sources of depth information have been briefly discussed in section 1.2. Among them, pictorial cues provide qualitative depth information (relative depth) rather than quantitative depth measures. Moreover, they involve complicated high-level reasoning. Therefore, it is difficult to formulate the cues and adopt them in computer vision system directly. Movement-produced cues have been used in finding depth. Examples of them are [KAPP87] [KOCH93]. However,

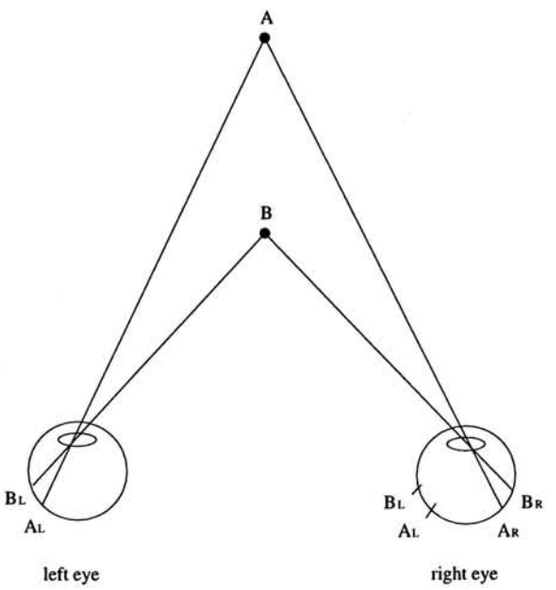


Figure 1.4: Disparity. Images of objects fall on different positions on our two retinae. The displacement (from A_L to A_R and from B_L to B_R) is known as disparity. Note that disparity is correlated with depth. Nearer objects produce greater disparities.

their applications are more restricted compared with depth from binocular disparity. In [GROS95], movement-produced cues (optical flow) is combined with binocular disparity in providing stereo vision. Oculomotor cues involves the sensation of eye muscles. Recently, researches on active vision make use of cues like vergence and focus [AHUJ93] [DAS95], which is similar to the oculomotor cues, however, they are used to a lesser extent than binocular disparity and cannot be used alone in reconstruction of real scene surfaces [AHUJ93]. Moreover, they exhibit certain shortcomings, such as the phenomena of *depth of focus* and *depth of field* for depth from focus. Depth from vergence faces similar problems with depth from stereo. However, depth from vergence only finds the depth of a single point called the point of fixation one at a time.

On the other hand, binocular disparity is the most important cue for depth in human depth perception[GOLD89]. It has also been widely made use of in computer stereo vision systems. A famous and classical example of them is the

Marr-Poggio algorithm [MARR79]. Depth from binocular disparity will be the main topic to be tackled in this thesis.

1.4 The Process of Stereo Vision

If only one 2-D image is present, the accurate depth information of a particular point in the image cannot be determined. i.e., the world coordinates (x_w,y_w,z_w) of the point cannot be found from its image point coordinates (x_i,y_i). This is because an image is in fact a 2-D projection of 3-D points onto an imaging plane. The mapping of the 3-D scenes to 2-D images is a many-to-one mapping. An image point (x_i,y_i) on a 2-D image may be the image of any world point lying on the line that passes through the points $(x_i,y_i,0)$ and (x_o,y_o,f), where x_o and y_o are the x-coordinate and y-coordinate of the center of the imaging lens and f is the distance between the lens and the imaging plane, called the focal length of the lens if the lens is focusing on a distant object.

The missing depth information can be obtained using stereo imaging techniques. This involves getting multiple images of the same object point from different view-points. When two images are used, it is called binocular stereo vision, so as to distinguish from trinocular stereo vision, in which three images are used.

1.4.1 Depth and Disparity

There are two types of imaging geometry for stereo cameras, namely the parallel axis stereo geometry and the nonparallel axis stereo geometry. Fig. 1.5 shows the parallel axis stereo geometry. In this geometry, the two imaging planes are on the same plane, the coordinate system of both cameras are perfectly aligned, differing only in the location of the origins. The relationship between disparity and depth will be explained using this stereo geometry. O_L is the origin of

the coordinate system on the left imaging plane. X_L and Y_L are the x and y coordinate axes of the left imaging plane. Similarity, O_R, X_R and Y_R represent the same things on the right imaging plane. The origin of the world coordinate, O_W, coincides with O_L. The x and y axes of the world coordinate system X_W and Y_W coincide with those of the left imaging plane. Z_W is the z-axis of the world coordinate.

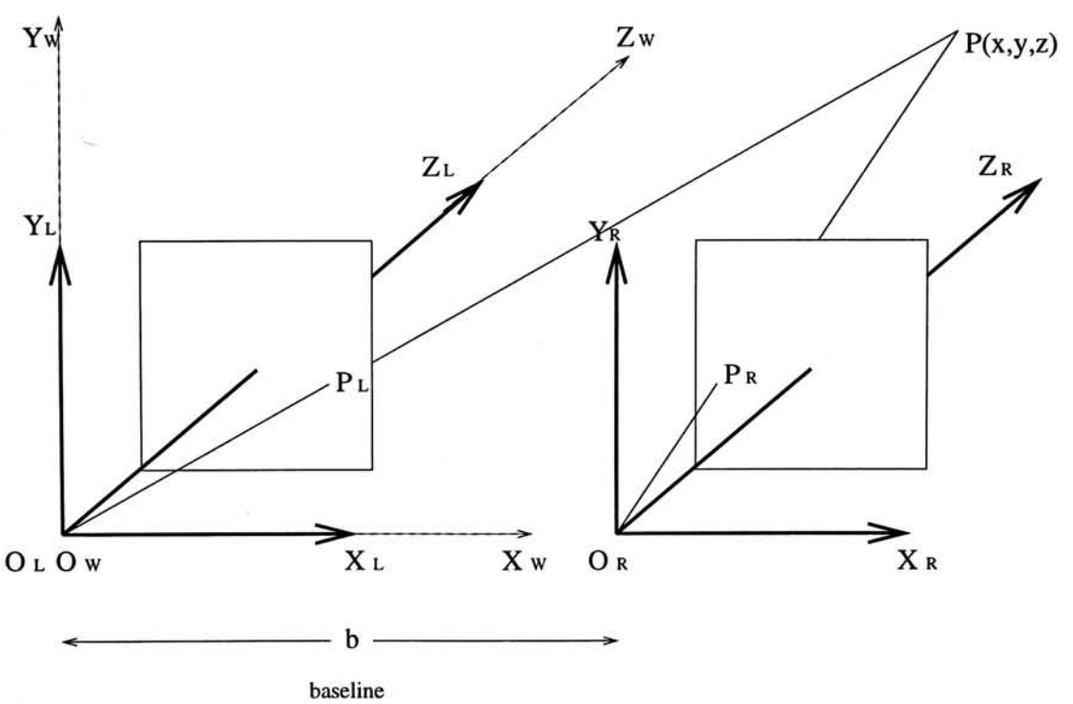


Figure 1.5: Parallel axis stereo geometry.

Fig. 1.6 shows a view perpendicular to the plane formed by the x and z axes. I_L and I_R are the left and right imaging plane respectively. C_L and C_R are the center of the left and right imaging lens respectively. b is the baseline, i,.e. the separation between the left and right imaging planes. Suppose there is a scene point $P(x,y,z)$ and that $P_L(x_l,y_l)$ and $P_R(x_r,y_r)$ are its left and right images. The disparity d is defined as $d = x_l - x_r$. By simple geometry, the world coordinates of the scene point P can be obtained as

$$x = \frac{bx_l}{d}, \qquad y = \frac{by_l}{d}, \qquad z = \frac{bf}{d}$$

where b is the baseline and f is the distance between the lens and the imaging plane.

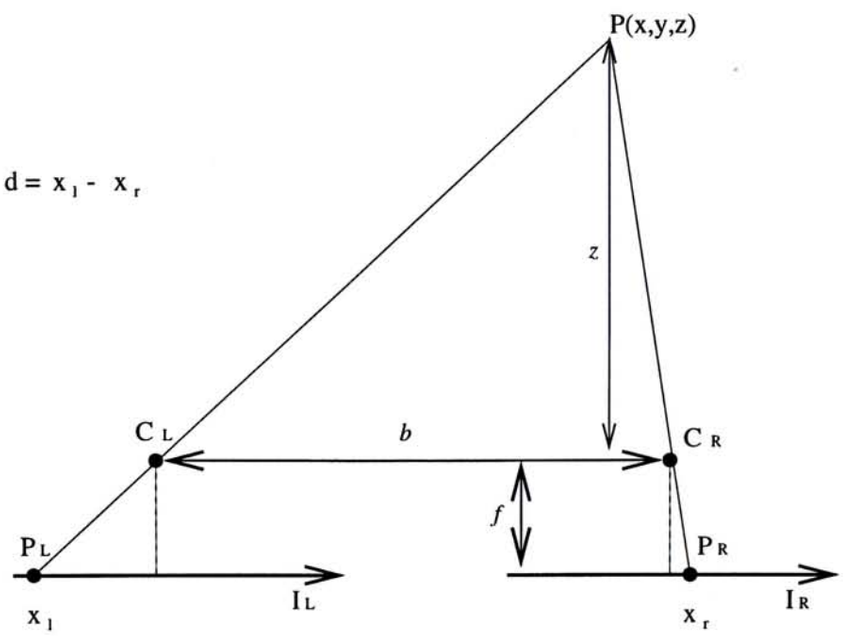


Figure 1.6: Figure showing relationship between depth and disparity.

1.4.2 The Stereo Correspondence Problem

The major steps in a typical computer stereo vision system are preprocessing, finding stereo correspondences and depth recovery (3-D structure determination). The most difficult step is the finding of stereo correspondences. It has been explained in section 1.4.1 that in general a scene point will have its images on both the left and the right imaging planes. For any image point on the left image, if its corresponding point on the right image can be found, the disparity, and hence the world coordinates, of the corresponding scene point can be found. This is called the stereo correspondence problem. Instead of matching the images pixel by pixel, different matching primitives are usually used. Matching primitives may be area patches, edge points, edge segments, zero-crossing patterns, etc.

Depending on what matching primitives are to be used in the step of stereo correspondence finding, the preprocessing step processes the images so as to prepare the matching primitives for matching, e.g. edge detection.

After the correspondences are found, the 3-D structure determination step reconstructs the 3-D coordinates of the scene points. See section 1.4.1 for details.

1.4.3 Parallel and Nonparallel Axis Stereo Geometry

According to [DHON89], stereo matching strategies can be differentiated in the broadest scene according to the primitives used for matching as well as the imaging geometry. Imaging geometry can be differentiated into parallel axis stereo geometry and nonparallel axis stereo geometry. The parallel axis stereo geometry has been introduced in section 1.4.1. Fig. 1.7 shows an example of the nonparallel axis stereo geometry. The imaging planes are not on the same plane and the optical axes of the cameras are not parallel. Conventionally, the parallel axis stereo geometry is used. However, because nonparallel axis stereo geometry has the advantage that it allows for a greater overlap of the left and right images, it has been used in some stereo systems. Examples are [AYAC87b], [AYAC87a] and [ITO86].

On the other hand, the disadvantages of nonparallel axis stereo geometry is two-folded. First, the epipolar lines are not parallel and in general not horizontal. Extra epipolar line computations become necessary before matching. Second, the definition of disparity becomes more complex and the 3-D reconstruction process requires a more general approach. Trading off the advantages and disadvantages, parallel axis stereo geometry will be used in the discussion hereafter and adopted in the matching algorithm explained in the later chapters.

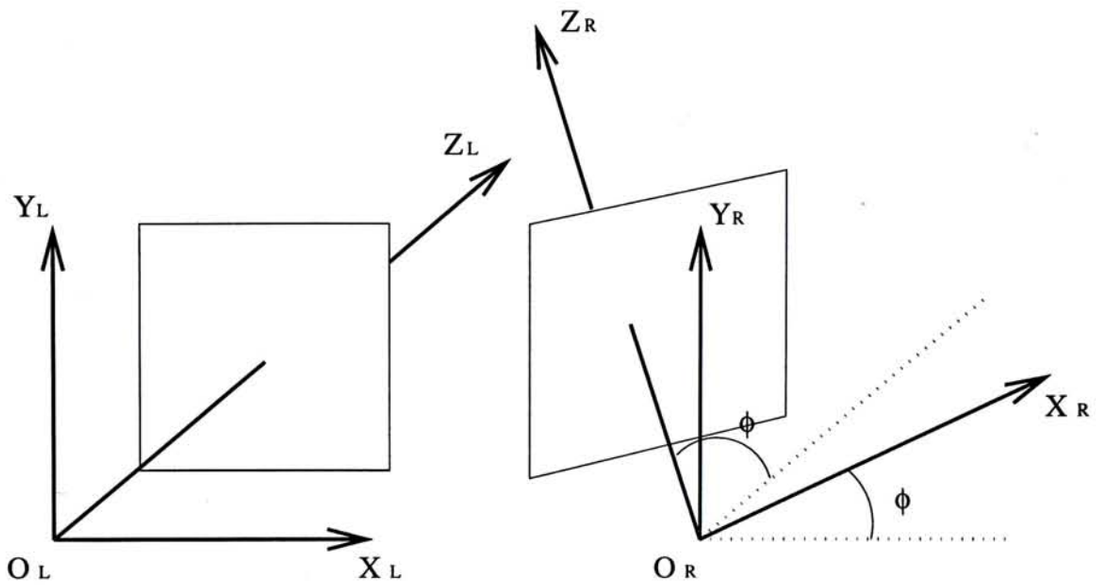


Figure 1.7: Nonparallel axis stereo geometry.

1.4.4 Feature-based and Area-based Stereo Matching

According to the matching primitives used, stereo matching strategies can be classified into two types, feature-based (edge-based) and area-based (region-based) matching. In feature-based matching, features like edge points, edge segments and zero-crossing patterns are used in the matching process. Most of the contemporary stereo algorithms are feature-based. In area-based matching, area patches are matched by cross-correlation between intensity patterns in the two images.

Because features can be characterized by their attributes (edge length, strength, orientation, etc.), feature-based matching can be performed using simple comparison of the attributes, and are faster than area-based matching. Moreover, they are more stable towards changes in contrast and ambient lighting. However, because feature-based matching only find the disparities of the features, they result in sparse disparity maps. Interpolation of disparities may be necessary to construct a dense disparity map, but it is not a straightforward process. Area-based matching results in dense disparity maps. However, if occluding

boundaries are present in the correlation window, the matches tend to be erroneous. This problem will be handled in the novel matching algorithm, which is an area-based one, presented in the later chapters.

1.4.5 Constraints

Due to their inverse nature, many vision problems are ill-posed. Stereo matching also possesses this ill-posed nature. The correspondence problem can be ambiguous. This is because an image point in one image may find more than one possible correspondences in the other image. This is known as the false target problem (fig. 1.8). In 1979, Marr and Poggio proposed a computational theory of depth perception [MARR79]. In their theory, the matching conditions between two binary images are represented by three rules :

1. Compatibility : the points that are regarded as in correspondence should be homologous, e.g. black dots can match only black dots.

2. Uniqueness : every point in one image can almost always (except for transparent objects) match no more than one point.

3. Continuity : the disparity of the matches varies smoothly almost everywhere over the image, except where depth discontinuities occur at surface boundaries.

The three rules have been the fundamental considerations in stereo matching. Other important constraints have also been proposed by researchers to facilitate search for correspondence and to eliminate false targets.

1. Epipolar Constraint

   For a given point in an image, its possible matches in the other image all lie on a line called the epipolar line. In the case of parallel axis stereo geometry, the epipolar lines are parallel and horizontal. This constraint

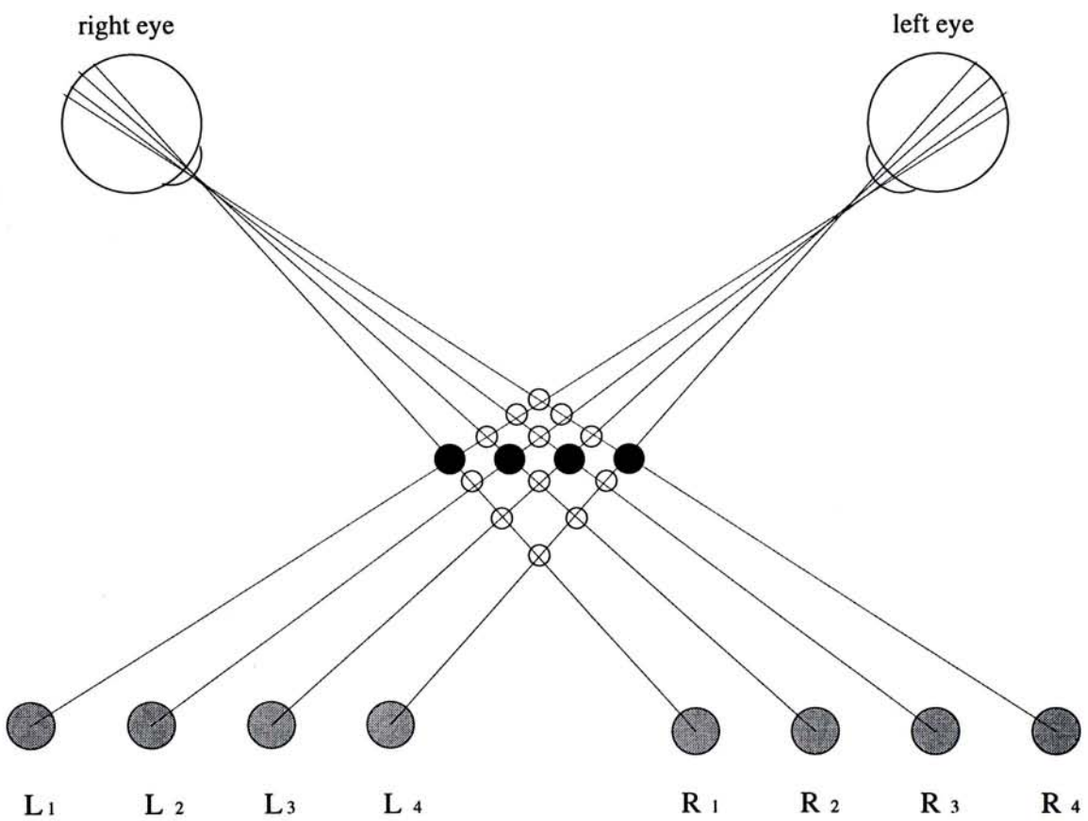


Figure 1.8: The false target problem. In this figure, L_1 to L_4 correspond to four perceived points in the left eye view, while R_1 to R_4 correspond to four perceived points in the right eye view. A correct match between the 4 pairs of points is represented by the filled circles in the middle, in which L_1 matches with R_1, L_2 matches with R_2 and so on. Incorrect matches are represented by the unfilled circles, e.g. L_1 matches with R_2, L_2 matches with R_3 and L_3 matches with R_4 result in perception of the row of false matches just above the row of correct matches in the figure. When the points are considered separately, each point on the left can match any point on the right. Without global consideration, such ambiguities cannot be resolved.

is very important in that it reduces the search for correspondences in 2-D images to 1-D. [BAKE81]

2. Ordering Constraint

   This is demonstrated in fig. 1.9. For a 3-D point M and its neighbor N, their projections in the two images should be in the same order (from left to right). Fig. 1.10 shows a case where the ordering constraint does not apply. Therefore, the ordering constraint is actually a constraint on the shape of the objects. It facilitates the assumption of simple surfaces instead of more complicated ones in case of ambiguity. [FAUG93]

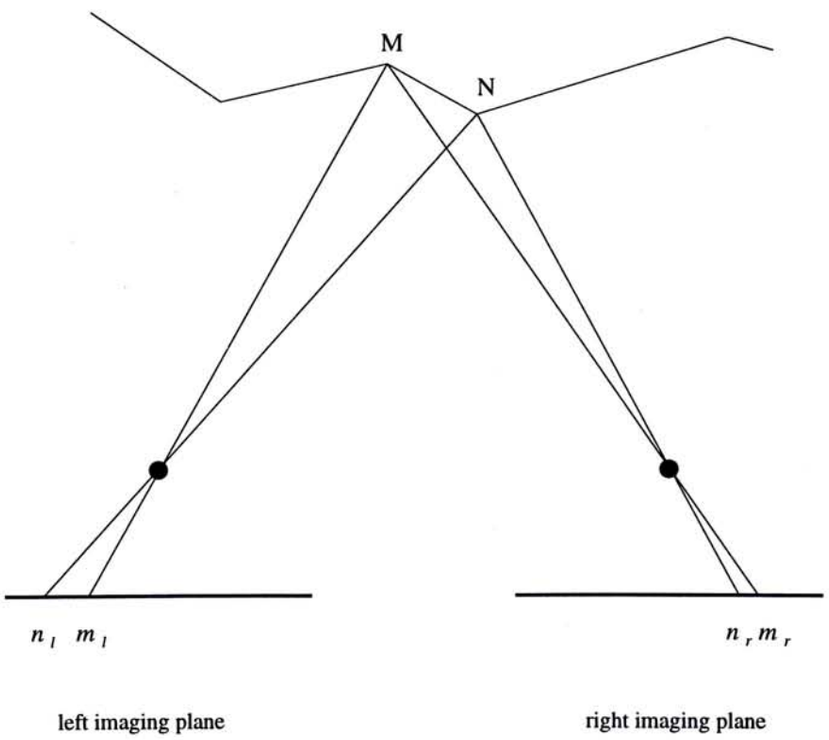


Figure 1.9: Ordering Constraint. Images of objects are in the same order on both imaging planes.

3. Disparity Gradient Limit

   A measure of disparity gradient of two points is defined as

$$DG = |\frac{d_1 - d_2}{w_1 - w_2}|$$

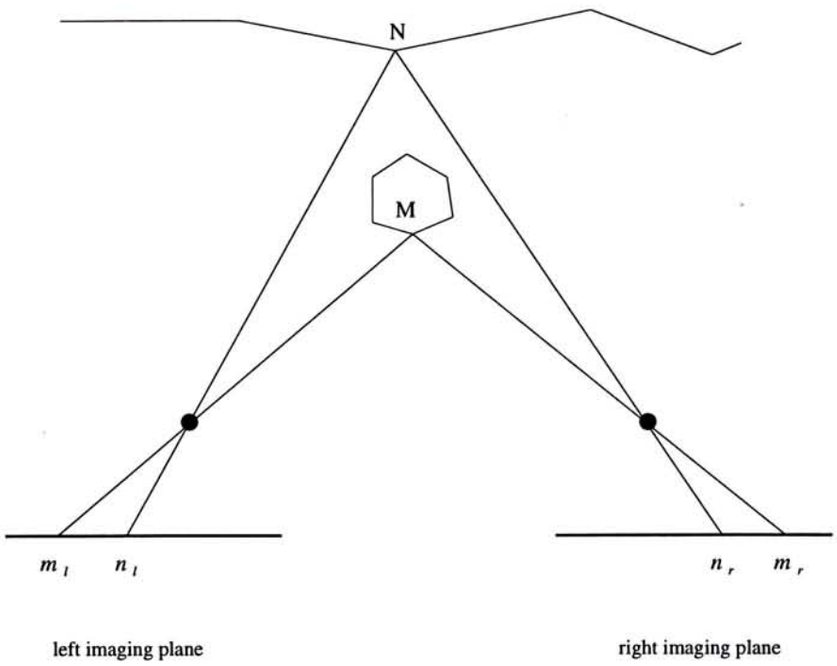


Figure 1.10: A case in which ordering constraint does not apply.

where w_i is the coordinates of two points on the cyclopean plane (an imaginary imaging plane parallel to the two real ones and in half-way between them, see fig. 1.11). Experiments in psychophysics have led people to conjecture that there is an upper bound of it in human perception. [POLL85]

4. Figural Continuity

   It is an extension of the continuity rules of Marr and Poggio requiring continuity of disparity along contours. It has been used in Grimson's computer implementation of Marr and Poggio's theory [GRIM85].

1.5 Organization of this thesis

This chapter gives a brief introduction to the problem of machine stereo vision. Chapter 2 presents some related previous work on the topic. Chapter 3 gives

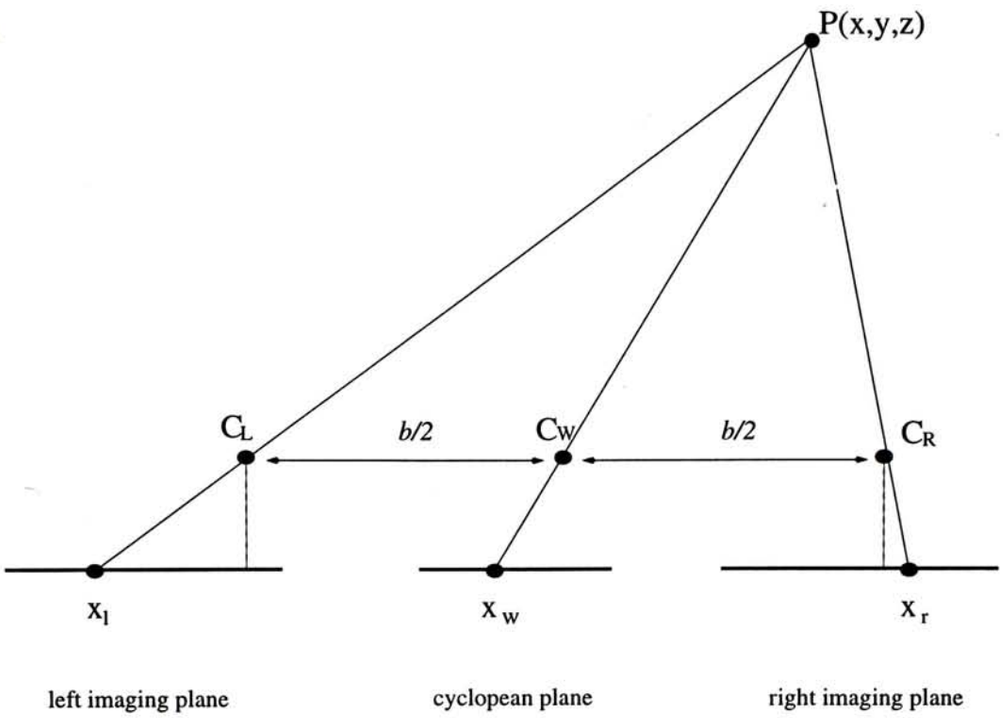


Figure 1.11: Cyclopean plane and disparity gradient.

an overview of a novel stereo matching method. Chapter 4 describe the pre-processing stage, i.e. edge detection and the extraction of matching primitives. Chapter 5 describe the matching stage, which is the main core of the matching method, and some related topics. Chapter 6 presents the Wavelet Representation of images and its use in a coarse-to-fine matching scheme. Chapter 7 gives some experimental results and analysis.

Chapter 2

Related Work

2.1 Marr and Poggio's Computational Theory

In 1979, Marr and Poggio proposed a computational theory of depth perception [MARR79]. They proposed that three steps are involved in measuring stereo disparity : (1) A particular location on a surface in the scene must be selected from one image; (2) the same location must be identified in the other image; (3) the disparity between the two corresponding locations must be measured. And they also formulated three matching constraints : compatibility, uniqueness and continuity (see section 1.4.5) and showed that the constraints can solve the false target problem. Before that, in 1960, Bela Julesz devised computer-generated random-dot stereograms. He also proposed that stereo matching is a cooperative process. Then until 1977 several algorithms are proposed by various researchers to solve the stereo vision problem and almost all of them are cooperative algorithms. According to Marr [MARR82](pp. 122), none of the algorithms was accompanied by an analysis of the underlying computational theory of the stereo matching problem (except [MARR76]) and therefore none of them are correct.

Grimson implemented the computational theory of Marr and Poggio and

addressed some implementation details [GRIM81] [GRIM85]. The intensity images are filtered by the Laplacian of Gaussian ($\nabla^2 G$) operator and zero-crossing points are detected in the images and grouped in 12 directional bins. Filters of three or four different sizes are used and the matching is a coarse-to-fine approach with disparities found at coarser resolutions used to guide searches at finer resolutions. [GRIM81] imposes a regional continuity check on disparity in case of ambiguous matches. If ambiguous matches are found at certain position, disparities in the neighborhood are taken into account. Later, [GRIM85] uses the figural continuity constraint of Mayhew and Frisby [MAYH81] that require continuity of disparity along contours instead of simple regional continuity.

2.2 Cooperative Methods

The earliest stereo algorithms proposed are cooperative methods. This results from Julesz's proposal that stereo matching is a cooperative process. The matching process is formulated as a parallel, interconnected network of processors with inhibitory and excitatory connections. As shown in fig. 2.1, the axes L_x and R_x represent the positions of descriptive elements in the left and right images. Every point on the plane contains a node in the network, and the node represents a possible match between the two positions represented by the two coordinates. It is easy to see that the diagonal lines are lines of equal disparities. The processor at each node performs the lowest-level matching processes between the two positions on the left and right image. The idea is that, with the effects of the excitory and inhibitory connections, the network will finally results in a state that all nodes representing correct matches will have a value 1, those representing incorrect matches contain 0.

One example of these algorithms is [MARR76]. It directly follows the Marr and Poggio's computational theory (section 2.1). Fig. 2.2 shows the connections

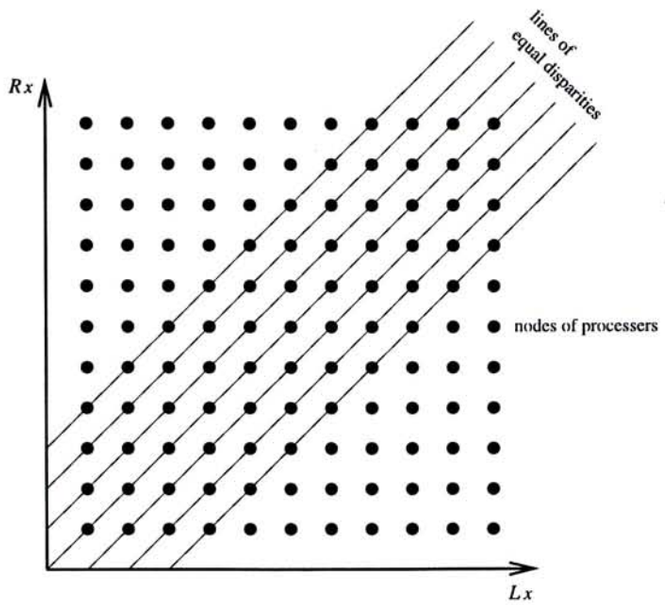


Figure 2.1: A representation of cooperative methods.

of the network. The solid vertical and horizontal lines represent lines of sight from the left and right image. The dashed diagonal lines represent lines of equal disparity. The compatibility constraint is carried out by the individual processors. The solid lines represent inhibitory connections. This carrys out the uniqueness constraint because only one correct match should happen on the same line of sight. The dashed lines represent excitory connections. This carrys out the continuity constraint, favoring matches that result in smooth disparity changes.

Some other researchers had proposed other cooperative methods with different excitory and inhibitory connections.

In more recent research, artificial neural networks and connectionist approach are still used to solve the stereo correspondence problem, examples are [MOUS94] [LEUN94] and [OR91].

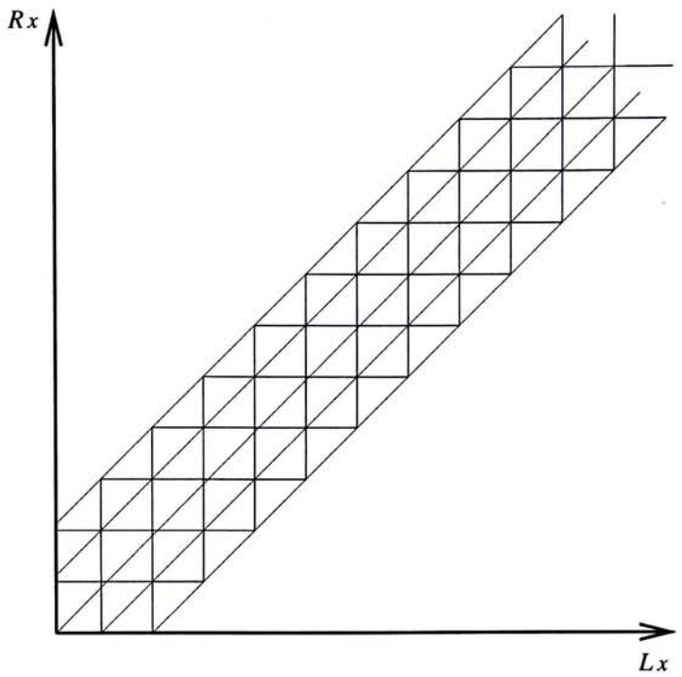


Figure 2.2: The Marr and Poggio's cooperative algorithm. The diagonal lines are excitory connections and the horizontal and vertical lines are inhibitory connections.

2.3 Dynamic Programming

The stereo correspondence problem can be treated as a problem of minimizing a cost function. The cost function minimization can then be solved by using dynamic programming. Two frequently referenced examples of using dynamic programming are [BAKE81] and [OHTA85].

Suppose parallel-axis geometry (or rectified images) is used so that the epipolar lines are the parallel horizontal scanlines and edge points are used as the matching primitives. Consider two corresponding scanlines on the left and right image. On each scanline, a number of edge points are identified and are numbered from left to right from 0 to $N-1$ on the left scanline and from 0 to $M-1$ on the right scanline. Then the problem can be represented by a two-dimensional grid as shown in fig. 2.3. The points on the grid represents possible matches. Point (i, j) means a candidate match of edge point i on the left scanline to edge

point j on the right scanline. The goal is then to find a sequence of match point from point $m_0 = (i, i')$ to point $m_e = (j, j')$, where (i, i') is the leftmost match and (j, j') is the rightmost match. By applying an ordering constraint on the sequences of match points (i.e. the sequences must be monotonic), the search space can be reduced considerably. A best sequence from m_0 to m_e can then be found using dynamic programming by finding the minimum cost $C(m)$ from m_0 to m_e recursively :

$$C(m) = \min_{p \ in \ V_m} (c(p, m) + C(p))$$

where V_m is the set of nearest neighbors of m before m, $c(i, j)$ is the cost from point i to j. The cost $c(i, j)$ can be based on similarity measure between features i and j, in which i is on the left image and j is on the right.

Baker and Binford [BAKE81] use an edge-based dynamic programming approach, in which each edge is treated as a doublet, with a left half-edge and a right half-edge. Matching is done using dynamic programming to match the half-edge on each scanline pair. A Cooperative process then follows to enforce the inter-scanline consistency using an edge continuity constraint, which is equal to applying a figural constraint. Finally, intensity-based matching is performed between intensity pixels from scanline intervals lying between the paired edges to yield a denser depth map.

Ohta and Kanade [OHTA85] proposed *stereo by intra- and inter-scanline search using dynamic programming.* Intra-scanline search means the matching between feature points on separate scanlines. The global consistency is established between scanlines by interscanline search. This imposes a consistency constraint among the matches found in intrascanline search. The problem is solved as a dynamic programming problem in 3-D search space instead of the previous 2-D space. The 3-D search space is a stack of the 2-D search spaces. The intra- and inter-scanline search proceed simultaneously, the matching score

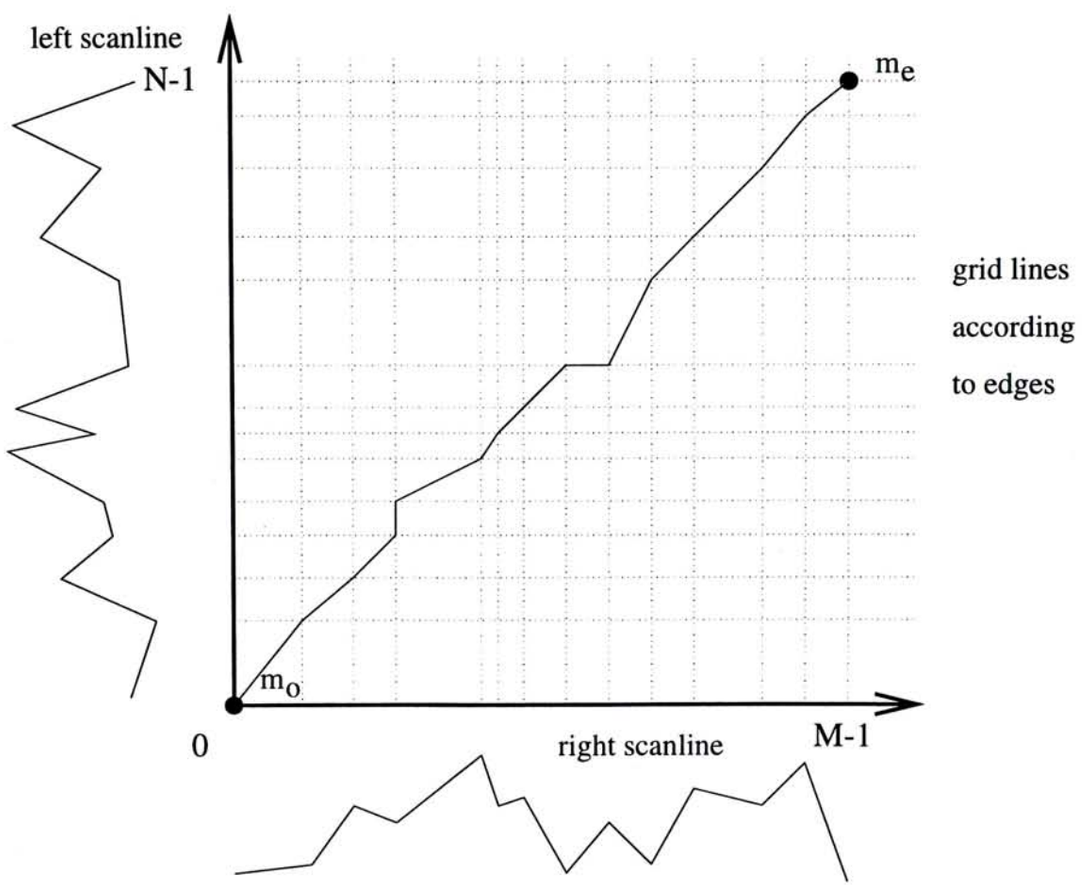


Figure 2.3: The stereo matching problem considered as a path finding problem on a plane. This plane represents a search between a pair of scanlines.

(intra-scanline) and consistency constraint (inter-scanline) cooperates in the process.

Li [LI94] used dynamic programming in Hough space instead of image space. Line-matching problem in image space is converted into peak point-matching problem in Hough space. i.e., dynamic programming is performed in Hough space using the peak points in Hough space as matching entities. It is claimed that figural continuity constraint is naturally embedded in the search and therefore complexity of the search process is significantly reduced and accuracy is improved.

2.4 Feature-based Methods

Image features such as edge pixels, edge segments, zero-crossing patterns, image contours and points selected by certain interest operators are commonly used in feature-based methods. Kim and Aggarwal [KIM85] uses *zero-crossing patterns* as matching features and introduces a relaxation method to find the best matches. 3 by 3 vertical zero-crossing patterns are used as the matching primitives. Horizontal zero-crossing patterns are not used because the search for matching takes place on the same scanline. The images are first smoothed to smooth out the noise and a 1-D Laplacian operator is applied to the smoothed images. They used two different methods to do this. The first method convolves the images with the Laplacian of Gaussian function, i.e.,

$$\nabla^2 G(x,y) = \frac{x^2 - \sigma^2}{\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)}$$

where

$$\nabla^2 = \partial^2/\partial x^2$$

$$G(x,y) = \sigma^2 e^{-(x^2+y^2)/(2\sigma^2)}$$

This is not a circular operator but a oriented operator because only vertical zero-crossings are interested. Another method uses a 3 by 3 averaging filter and apply 1-D Laplacian to the smoothed image

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

$(a) averaging filter \quad (b) 1-D Laplacian operator$

These two operations can be combined as a single convolution of the image with the 1-D Laplacian of an averaging filter :

$$\frac{1}{9}\begin{bmatrix} 1 & -1 & 0 & -1 & 1 \\ 1 & -1 & 0 & -1 & 1 \\ 1 & -1 & 0 & -1 & 1 \end{bmatrix}$$

In this way, 9 possible zero-crossing patterns are possible (fig. 2.4), each of them is assigned a number. The left and right images are preprocessed using one of the above operators. Then for each zero-crossing point in the left image, initial weights are assigned to every candidate point in the right image according to similarity of zero-crossing patterns and the difference in intensity gradients. Then a relaxation process is applied to find the best match. The relaxation process is based on the figural continuity, disparity continuity and certainty of matches of the zero-crossing patterns. The search is limited to a maximum possible disparity value (d_{max}), which is determined by the application environment.

Horaud and Skordas [HORA89] use linear edge segments as matching primitives. Each segment is characterized by its position and orientation. Moreover, the relationships between nearby segments are taken into consideration. A relational graph is built for each image to encode the relationships between nearby segments. A correspondence graph is then built such that each node represents

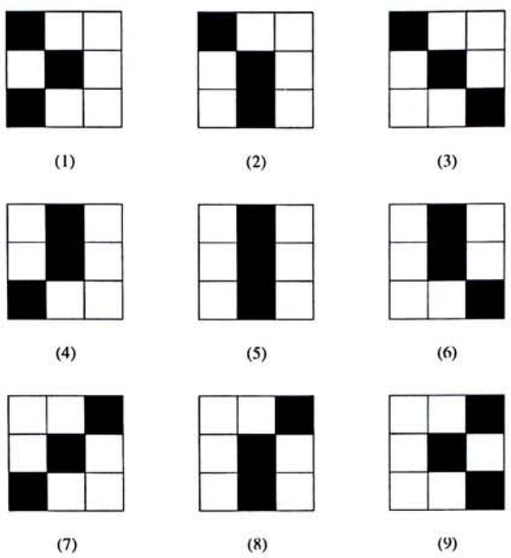


Figure 2.4: Zero-crossing patterns used by Kim and Aggarwal.

a possible correspondence. Arcs in this graph represent compatible correspondences on the basis of segment relationships. The stereo matching problem then is cast to searching for sets of mutually compatible nodes in the correspondence graph. A benefit function is defined on the similarity between the segments in terms of contrast, length and orientation. The maximal clique which maximize the benefit function is chosen as the best match.

2.5 Area-based Methods

In area-based matching, area patches are matched by cross-correlation between intensity patterns in the two images. A classical example is Moravec's robot cart [MORA80]. Moravec developed a vision system for a robot cart that made use of stereo vision in path planning. Points of interest are identified in each image by the Moravec interest operator. For each point of interest, the target image is searched at various resolutions starting from the coarsest. At each resolution the point of interest is matched with the target image using correlation of intensity as matching measure. The position that yields the highest correlation score

is enlarged to the next finer level of resolution. The process continues until the finest resolution is reached. The interest operator is described in [MORA80]. Moravec used 4 × 4 windows. Sums of squares of differences of pixels adjacent in each of four directions (horizontal, vertical and two diagonals) over each window are calculated, and the window's interest measure is the minimum of the four sums. The interest operator tried to locate features like angles in an image and at the same time reject simple edges. The interest operator is as follows :

$$m_1 = \sum_{i=1}^{4}\sum_{j=1}^{3}(P(i,j) - P(i,j+1))^2$$

$$m_2 = \sum_{i=1}^{3}\sum_{j=1}^{4}(P(i,j) - P(i+1,j))^2$$

$$m_3 = \sum_{i=2}^{4}\sum_{j=2}^{4}(P(i,j) - P(i-1,j-1))^2$$

$$m_4 = \sum_{i=1}^{3}\sum_{j=1}^{3}(P(i,j) - P(i+1,j+1))^2$$

the m_1, m_2, m_3, m_4 measure directional variance in horizontal, vertical, and the two diagonals respectively. The window's interest measure is the minimum of the four.

Images taken from nine cameras are used in the matching process to provide redundancy of data. The same correlation process is applied to the nine images two at a time. Therefore for each point there are C_2^9 possible disparities. The disparities and correlation coefficients are used to calculate a confidence measure of the final result.

Gennery [GENN80] proposed a high-resolution correlator. Sub-pixel accuracy is achieved with statistics of noise in the images, brightness and contrast adjustment and interpolation. Apart from finding the match point, the correlator also return a probability value, which is an estimate of the accuracy of the match based on statistics of noise in the image intensity. The images are divided

into fixed-size and non-overlapping square areas. Matching is done column by column from left to right. The search is restricted to a distance range, which is determined by using a priori information about the scene. The stereo disparities are allowed to vary in an arbitrary way over the picture, subject to only mild local continuity constraints.

Toh and Forrest [TOH90] proposed an edge and shading encoding scheme for images and demonstrated its use in stereo matching. Their method encodes image as piecewise surface shading separated by edges. The surface shading is encoded as smooth polynomials. In stereo matching, the matching is done by comparing the coefficients of the polynomials across corresponding scanlines. This can be regarded as an area-based matching without doing cross-correlation because the polynomial coefficients are regarded as approximate representations of area patches.

Cohen, Sander and Gagalowicz [COHE89] gives a method that performs stereo matching using regions as matching primitives. The implementation maintains a hierarchy of segmented regions in each image. Each level in the hierarchy corresponds to analysis at different scales. Instead of doing the segmentation process before matching, the two processes are treated as related processes. The two processes proceed simultaneously in a cooperative fashion in that partial matching results are used to enhance the segmentation process.

Cochran and Medioni [COCH89] gives an area-based method that use the entire images in the cross-correlation process. First, the cross-correlation for the entire images limited to a certain disparity range is generated. The cross-correlation is performed both from left to right image and from right to left. A dense disparity map is then obtained by selecting the disparity values according to 4 criteria with priority.

1. Cross-correlation peaks on each view that is at least 50

2. Cross-correlation peaks which are exactly in agreement by both view and

which adjoin existing agreed-upon points.

3. Cross-correlation peaks which are within a bounded agreement from both views and which adjoin existing agreed-upon points.

4. Non-peak points which are within a bounded agreement from both views and which adjoin existing points.

They argued that the disparity map obtained this way is a good estimate of the disparity, but is *blurred* across the depth discontinuity edges because of the cross-correlation. Further refinement of the disparity map is carried out after edge detection. No independent feature-based matching process is carried out with the edgels. The edgels is assigned with the disparity values obtained from the cross-correlation process. The disparity map is then smoothed, keeping the disparity at the edgels fixed, to remove the blurring effect across depth discontinuity.

Chapter 3

Overview of the Method

This and the following chapters describe an area-based stereo matching method. This chapter gives an overview and the main considerations behind the methods. Its components will be described and explained in the following chapters.

Area-patches are commonly used in area-based matching. However, in this method, horizontal intensity line segments are used as the matching primitives. Edges are detected in the left and right images and then horizontal line segments that are enclosed by two edge points on the horizontal scan-lines of the raw intensity images are extracted (fig. 3.1). This is effectively an area-based matching approach because most of the pixels in the intensity images are used. The horizontal intensity line segments can be considered as area-patches of 1-pixel width and variable length. Moreover, the method results in dense disparity maps, which is a characteristic of area-based matching.

Edges are not matched directly in this method, but they are important in extracting the horizontal intensity line segments. Therefore, the choice of edge-detector is also important to the final result.

The imaging geometry is a parallel axis geometry. (see section 1.4.3) Non-parallel axis geometry has the advantage that it allows for a greater overlap of the left and right images. However, the extra computations necessary to find

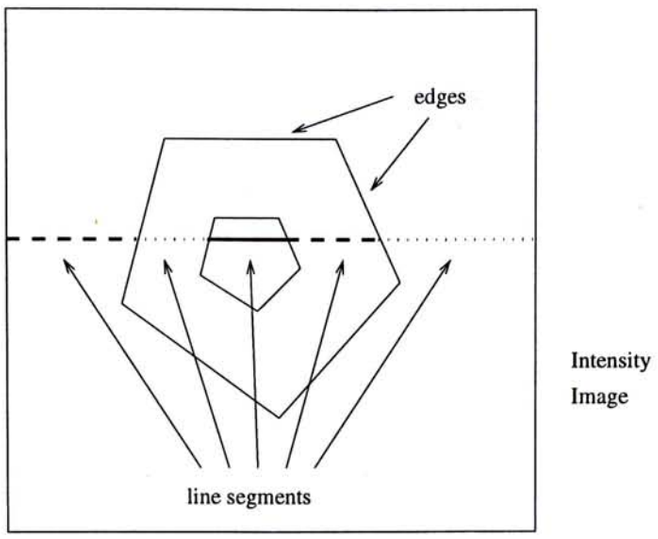


Figure 3.1: Illustration of how horizontal intensity line segments are extracted between edge pixels.

the epipolar lines or rectify the images out-weights its advantage. Therefore, we adhere to the simpler parallel axis geometry.

3.1 Considerations

The choice of using these extracted horizontal line segments as units of matching is made mainly because of the following considerations:

Dense Disparity Map The first consideration is a dense disparity map. Feature-based matchings are generally more accurate because they are less sensitive to noise and that area-based matchings do not work well at occluding boundaries. However, features are generally sparse within images, resulting in sparse disparity maps. On the other hand, area-based matchings produce less accurate but dense disparity maps. In our method, occluding boundaries are taken care of because of its choice of matching primitives. Moreover, the sparse

disparity maps produced by feature-based matchings usually require further processing until they are useful. Interpolation of disparity map is necessary, which is not straight-forward. Dense disparity maps produced by area-based matchings may also require further processing to fine-tune the disparity values. It is nevertheless less complicated.

Segments and Windows Second, the choice of using horizontal intensity line segments in effect dynamically choose suitable window sizes for matching different parts of the images according to the image characteristics in different parts of the images. Generally, in area-based matching, windows are used to divide the intensity images into units of matching instead of matching the pixels individually. The classical area-based matching methods used fix-sized windows (e.g. [MORA80] [GENN80]). On the choice of window-size, there is a dilemma: Large windows are more robust but they only give coarse disparity values because of the averaging property of cross-correlation. On the other hand, small windows result in finer disparity values but are sensitive to noise and break down at repetitive features, this is because small windows may not cover enough intensity variation for unique match. In this method, the length of the line segments plays the role of the size of the windows. The length of the line segments varies according to the position of edges that bound the area to which the particular line segment belongs. This effectively helps to choose a best window size for that particular area in the image being matched. Further explanation is presented in chapter 4.

Epipolar Constraint Third, based on the epipolar constraint (section 1.4.5), for any point on the left image, its correspondence point on the right image lies on the epipolar line. When parallel-axis stereo geometry is used, epipolar lines are horizontal scanlines. Therefore, for any point on the left image, its correspondence point on the right image lies on the same horizontal scanline in the

right image. This reduce the 2-D search into a 1-D search. Even if the images are slightly shifted vertically due to incorrect calibration, the matching should, to a certain extend, be fault-tolerant because there is generally coherence between adjacent rows of pixels unless at edge positions. Therefore, horizontal line segments of single pixel width can be used in the matching process without considering the vertical disparity continuity in the first step. The vertical disparity continuity can be considered and is helpful in smoothing the disparity map after the matching process.

3.2 Brief Description of the Method

The original intensity images is first passed to an edge detector. The Canny edge detector is used in this implementation. The edge detector gives distinct edge points after certain thresholding, i.e. a pixel is either an edge point or not, with no intermediate value. Then, for every horizontal scanline in the left image, edge points are located. The row of pixels bounded by two edge points, or bounded by the an edge point and the left or right boundary of the image are extracted as the horizontal intensity line segments (the matching primitives used in this method). Edge points are not included in the line segments. These will be further explained in chapter 4.

The matching process involves an area-based matching using the horizontal line segments. Cases of inclined surfaces and occlusions are handled. The area-based matching is done by computing a similarity measure between each line segments extracted from the source image and its candidate segments in the other image. The *sum of normalized difference* is used as the similarity measure. Refinement of the disparity map is then carried out to give the final disparity map. Further details are discussed in chapter 5.

The matching process can be enhanced using a coarse-to-fine approach. This

is described in chapter 6.

Fig. 3.2 depicts the relationship between the different stages.

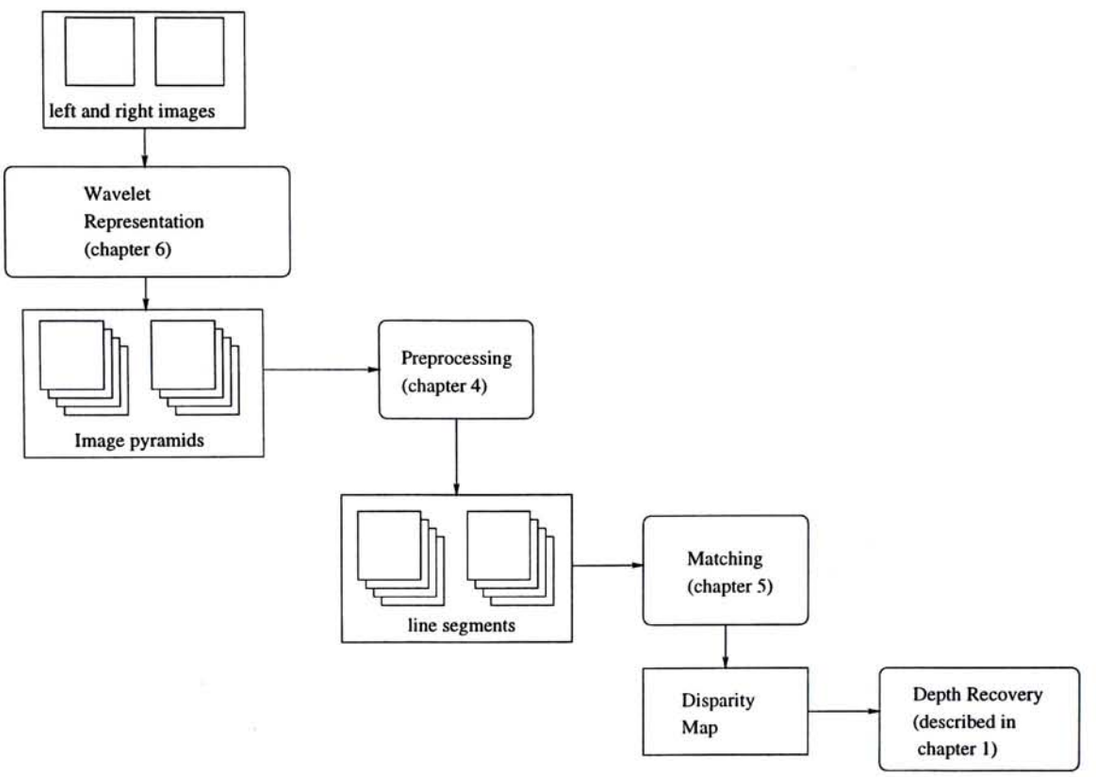


Figure 3.2: Relationship between stages.

Chapter 4

Preprocessing of Images

Preprocessing of images is an important component of the whole stereo matching method. The preprocessing stage segmentates the images and extract the matching primitives to prepare for further processing. In this method, horizontal intensity line segments bounded by edge pixels are used as the matching primitives. Preprocessing involves edge detection and the extraction of the horizontal intensity line segments.

4.1 Edge Detection

An edge is the boundary between two regions with relatively distinct gray-level properties. Basically, the idea underlying most edge-detection techniques is the computation of a local derivative operator. Fig. 4.1 gives an illustration of the idea. The second derivative has a zero-crossing at the midpoint of a transition in gray level. The midpoint of the transition locates the position of the edge and the magnitude of the zero crossing can be a measure of the strength of the edge. A similar argument applies to an edge of any orientation in an image.

Many different edge detectors have been proposed. Most of them can be classified into three main categories [BALL82]:

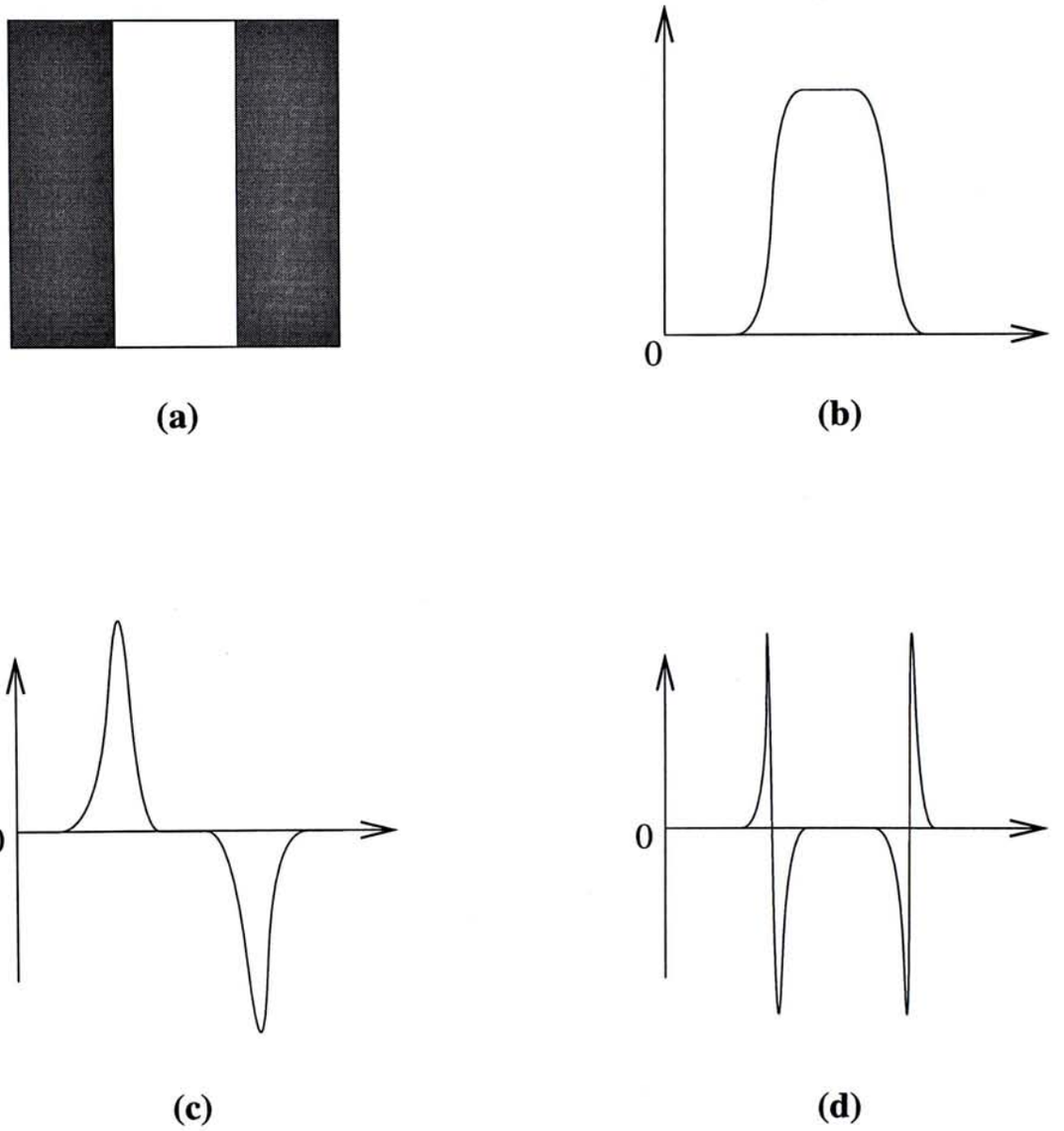


Figure 4.1: Edge detection by local derivative operator. (a) image of a light stripe on a dark background; (b) profile of a horizontal line; (c) first derivative of the horizontal line; (d) second derivative of the horizontal line.

1. Operators that approximate certain mathematical derivative operators;

2. Operators that involve convolution of the image with a set of templates tuned to different orientations; and

3. Operators that fit local gray-level intensity values surrounding a point with (edge) surface models and extract edge parameters from the model.

Although different edge detectors have been proposed to serve the edge detection purposes. They have different properties and produce edge maps that differs in details. A choice of operator can be made based on the characteristics of the particular matching algorithm. For example, the Laplacian of Gaussian ($\nabla^2 G$) operator has widely been used in many edge-based matching methods. However, the Canny edge detector [CANN86] is more suitable in this matching method and is used in the preprocessing phase.

4.1.1 The Laplacian of Gaussian ($\nabla^2 G$) operator

The Laplacian of Gaussian ($\nabla^2 G$) operator has been used widely for edge detection. The Laplacian of a 2-D function $f(x, y)$ is a second-order derivative defined as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

the Gaussian part (G) stands for the two dimensional Gaussian distribution

$$G(x, y) = e^{-\frac{x^2+y^2}{2\pi\sigma^2}}$$

with standard derivation σ.

The Laplacian of Gaussian operator may be expressed in terms of the radial distance r from the origin by the formula:

$$\nabla^2 G(r) = \frac{-1}{\pi\sigma^4}\left(1 - \frac{r^2}{2\sigma^2}\right) exp\left(\frac{-r^2}{2\sigma^2}\right)$$

The shape of the operator is a circularly symmetric inverted Mexican-hat-shape (fig. 4.2). The best engineering approximation to $\nabla^2 G$ is the difference of two Gaussians (DOG), when the ratio of the standard deviations of the inhibitory Gaussians to that of the excitory one is about 1 : 1.6 [MARR82]. Fig. 4.3 shows an example of an image and the effects of applying the DOG operator.

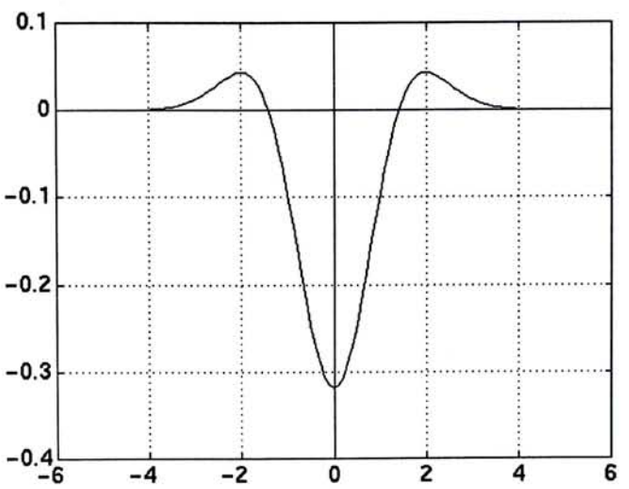
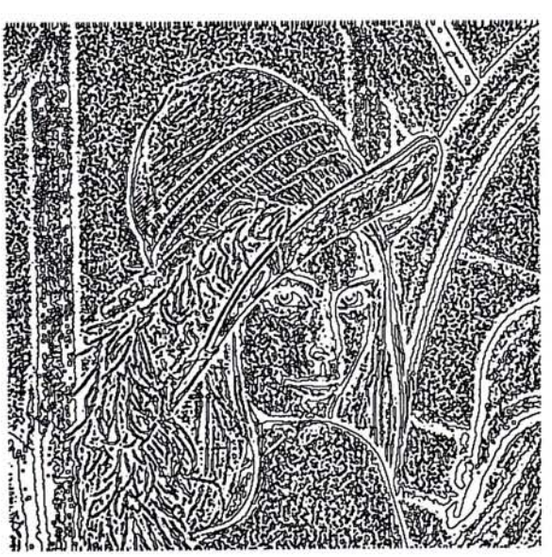


Figure 4.2: The Laplacian of Gaussian shown as a one-dimensional function ($\sigma = 1$)

Marr and Hildreth [MARR80] argued that an edge detector should be capable of being tuned to act at any desired scale, so that large filters can be used to detect blurry shadow edges, and small ones can be used to detect sharply focused fine details in the image. They also argued that the most satisfactory operator fulfilling this is the filter $\nabla^2 G$. The argument is that the Laplacian part is a differential operator, taking the second derivative of images. The Gaussian part is used to blur the images, effectively wiping out all structure at scales much smaller than the space constant σ of the Gaussian. The Gaussian distribution has the desirable characteristic of being smooth and localized in both the spatial and frequency domains. It is, therefore, least likely to introduce any changes that were not present in the original images. This is the reason why it is used to

(a)



(b)

(c)

Figure 4.3: The Lena image. (a) original image, 512x512x256G; (b) zero-crossings after applying the DOG operator; (c) only the highest 30% of pixels (in terms of magnitude of zero-crossings) is shown.

blur the images instead of using, say, a cylindrical pill-box function. The Marr-Hildreth edge detector [MARR80] is an approximation of the $\nabla^2 G$ operator. It convolves a mask approximating the $\nabla^2 G$ function over the entire image and labels the zero-crossings of the convolution output as edge points. The $\nabla^2 G$ operator or the Marr-Hildreth edge detector or their simplified forms has been widely used by many edge-based matching algorithms (e.g. [GRIM81], [GRIM85], [KIM85], [OR91]).

4.1.2 The Canny edge detector

As can be depicted from fig. 4.1, instead of detecting edges from zero-crossings of the second derivative, we can also use the extrema of the first derivative. This give rise to a family of edge detectors based on the detection of extrema in the output of the convolution of the image with an impulse response $h(x)$. The simplest of them is the difference of boxes operator [ROSE71] (fig. 4.4). A better one is the first derivative of Gaussian (FDG) operator :

$$h(x) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

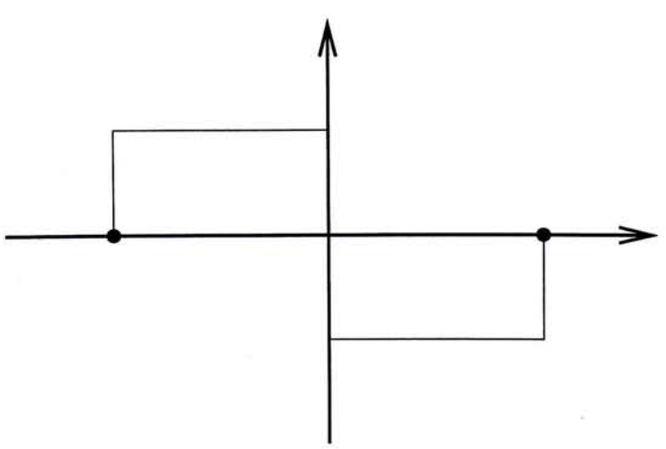


Figure 4.4: The Difference of boxes operator

The idea is similar to that of the Laplacian of Gaussian ($\nabla^2 G$), in which the Gaussian part is used to smooth the original image without introducing high-frequency changes, the first derivative is used to detect the edge.

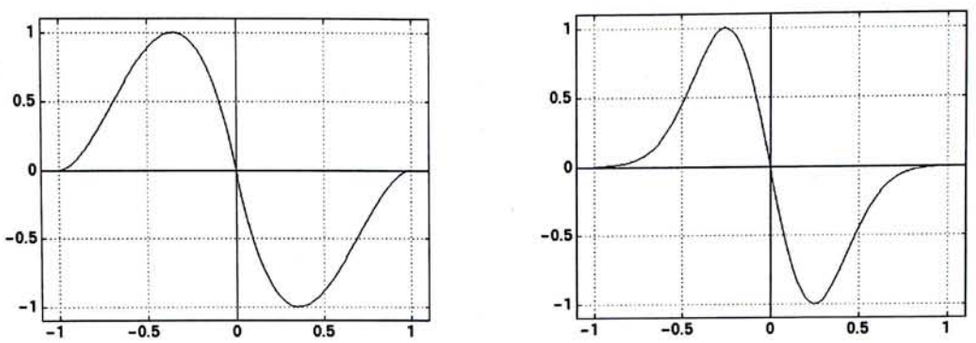
Canny [CANN86] proposed an optimal detector with well-stated goals of maximizing detection and localization, at the same time eliminating multiple responses. Canny formulated these criteria in mathematical forms and find the optimal operator by numerical optimization. The filter is presented in the mathematic form :

$$h(x) = a_1 e^{\alpha x} \sin \omega x + a_2 e^{\alpha x} \cos \omega x + a_3 e^{-\alpha x} \sin \omega x + a_4 e^{-\alpha x} \cos \omega x + c$$

for x in $[-W, 0]$, and circularly symmetric in $[0, W]$, where W is the spatial extent of the impulse response.

Canny's optimal operator is close to the FDG operator (fig. 4.5). Canny claimed that the performance of FDG is about 20 percent worse than Canny's operator in terms of detection and localization and about 10 percent in terms of multiple responses [CANN86]. A scheme of thresholding based on estimation of noise in the image intensity is also proposed. Two thresholds are computed, namely the *low* and *high* thresholds. All pixels with values above *high* are considered edges. All pixels with values below *low* are eliminated and all pixels with values between *low* and *high* are considered edges if they can be connected to a pixel above *high* through a chain of pixels above *low* . This thresholding scheme further improve the performance of the operator.

In experiments, when compared with the Difference of Gaussian (DOG), the Canny edge detector produce better edge map, the edge pixels are more connected and conform to the figural continuity of the image (fig. 4.6). Therefore, it is more suitable in this application, in which edge pixels are used to extract horizontal intensity line segments for matching. If the edge pixels are more connected and conform to the figural continuity of the image, the line segments extracted are thus more coherent among segments and conform to the figural



(a) (b)

Figure 4.5: (a) The Canny operator; (b) The first derivative of Gaussian (FDG); the figures are scaled to approximate size for comparison.

continuity of the image. This also facilitates the refinement of the final disparity map when figural continuity is taken into account. The Canny edge detector is used in the implementation of the stereo matching method presented in this thesis.

4.2 Extraction of Horizontal Line Segments for Matching

Both the left and right images are passed to the Canny edge detector. The Canny edge detector gives distinct edge pixels after its scheme of thresholding, i.e. a pixel is either an edge point or not. Strength of the edge pixels is not interested. For each horizontal row of pixels (scanline), edge points are located. Horizontal line segments are then extracted from the row of pixels such that the extracted line segments contain no edge points and are either bounded by two edge points or by an edge point and the left or right boundary of the image. Effectively, the edge pixels guide the extraction of the line segments. The extracted horizontal line segments are the matching primitives used in this

(a)

(b)

(c)

Figure 4.6: Comparison of output of edge detectors; (a) original Lena image 512x512x256G; (b) output of the Canny edge detector; (c) output of the Difference of Gaussian (DOG); in (b) and (c), the edge pixels count about 4.7% of the entire image.

matching method. Therefore, the edge pixels effectively guide the matching process. The methodology behind is based on the assumptions that:

1. Edges are likely to be points of disparity changes. So, they should not be included in the middle of a line segment to be matched. On the other hand, two area separated by an edge are better matched separately as it is more likely to have disparity changes between them. These factors favour the use of these line segments without edges as matching primitives.

2. The horizontal line segments extracted are then without abrupt change in intensity. It is likely that the disparities of all the points on the same line segment are the same or they vary only smoothly. Therefore, it is easier to decide the disparity values within a single line segment. This will be described in chapter 5.

3. When using cross-correlation techniques (or similar measures), abrupt change in intensity are likely to produce error. By excluding edges from the matching primitive, matching should be more accurate.

As explained in section 3.1, The choice of using these horizontal intensity line segments in effect dynamically choose suitable window sizes for matching different parts of the images according to the image characteristics in different parts of the image. This also speaks in favour of using these segments in cross-correlation processes.

Chapter 5

The Matching Process

The matching stage involves the area-based matching using the horizontal intensity line segments extracted in the way described in section 4.2. A novel matching scheme is designed for using these horizontal intensity line segments as matching primitives. Occlusion detection and refinement of disparity map are also presented.

5.1 Reducing the Search Space

In parallel-axis stereo geometry, epipolar lines are the horizontal scanlines. Assuming epipolar constraint, each scanline is matched only with its corresponding epipolar line, i.e. the same row of scanline on the other image. After the preprocessing stage (chapter 4), horizontal intensity segments are extracted according to edges detected. Suppose there are M segments on the left image, and N segments on the right image. There are altogether $M \times N$ possible matches. However, only a little part of them are correct and effort can be saved by eliminating the unlikely matches.

Maximum possible disparity value First, a maximum possible disparity value d_{max} can be assumed for every pair of stereo images. Then, for every

segment in the source image (left or right), its candidate segments are limited to those within displacement of $2 \times d_{max} + 1$ pixels. [KIM85] suggests that the d_{max} value can be determined for each application environment. We can first estimate the minimum distance D_{min} of the objects from the cameras. Once the minimum distance is known for one particular application environment, the d_{max} value can be found from the parameters of the imaging geometry:

$$d_{max} = \frac{Bf}{D_{min}}$$

, where B is the length of the baseline and f is the focal length of the cameras.

Segments against sliding windows With the use of horizontal intensity line segments as units of matching, the search space is small because there are usually only a few segments within the disparity range. When we use, for example, a fixed size sliding window as the unit of matching, there are $2 \times d_{max} + 1$ windows within the allowed disparity to be matched against if we slide the window at 1-pixel intervals. However, when we use the horizontal intensity line segments as units of matching, we do not need to slide the segments over one another. This is because we have made use of the information of edge positions when extracting the segments. The start and end of the segments correspond to positions of edges. When we compare two segments, we do not need to slide them over one another because there should be no sharp edge within the segments themselves, the start and end of segments should not be matched to the middle of another. Therefore, the use of segments as matching primitives help reduce the search space, in comparison with using sliding windows. There are exception cases when there are partial occlusions. In such cases, we cannot align both end of two segments. Partial occlusions will be elaborated in section 5.4.

Limit on difference of length of segments Furthermore, among the candidate segments lying within the allowable disparity region, those with a very

different length from that of the source segment can be rejected. It is further explained in section 5.3.

5.2 Similarity Measure

In this stereo matching method, horizontal intensity line segments are the matching primitives. For two segments (one from the left image and the other from the right one) to be matched, a similarity measure between the segments is computed to decide whether the segments are similar to each other. Each segment in the source image has a corresponding set of candidate segments in the destination image. Similarity measure should be performed between a source segment with every one of its candidate segments. The one among the candidate segments with greatest score in the similarity measure is made the correspondence (match) of the source segment. The established correspondences are used to construct the preliminary disparity map.

Cross-correlation A traditional similarity measure is the *cross-correlation.* Suppose a 2-D matching window($w(x,y)$ of size $J \times K$) and a 2-D image ($f(x,y)$ of size $M \times N$) are to be compared, the cross-correlation between $w(x,y)$ and $f(x,y)$ is :

$$c(s,t) = \sum_x \sum_y f(x,y)w(x-s, y-t)$$

where $s = 0,1,2,\ldots,M-1; t = 0,1,2,\ldots,N-1$, and the summation is taken over the image region where w and f overlap, and it is assumed that the origin of $f(x,y)$ is at its top left and the origin of $w(x,y)$ is at its center. The maximum value of $c(s,t)$ indicates the position where $w(x,y)$ best matches $f(x,y)$.

Correlation coefficient The cross-correlation above-mentioned has the disadvantage of being sensitive to changes in the image intensity. *Correlation coefficient* can be used to overcome this problem. It is defined as

$$\tau(s,t) = \frac{\sum_x \sum_y [f(x,y) - \bar{f}(x,y)][w(x-s,y-t) - \bar{w}]}{\left\{\sum_x \sum_y [f(x,y) - \bar{f}(x,y)]^2 + \sum_x \sum_y [w(x-s,y-t) - \bar{w}]^2\right\}^{1/2}}$$

where $s = 0,1,2,\ldots,M-1; t = 0,1,2,\ldots,N-1$, $\bar{w}$ is the average value of pixels in $w(x,y)$, $\bar{f}(x,y)$ is the average value of $f(x,y)$ in the region overlapping with w, and the summations are taken over the image region where w and f overlap. This is in fact the cross-correlation function normalized for intensity changes. Again, the maximum value indicates the position where $w(x,y)$ best matches $f(x,y)$.

Sum of normalized differences *Sum of normalized differences* can also be used :

$$d(s,t) = \sum_x \sum_y \left| \frac{f(x,y)}{\bar{f}(x,y)} - \frac{w(x-s,y-t)}{\bar{w}} \right|$$

, it is also normalized for intensity changes. It is easier to implement and consumes less computation than the correlation coefficient. This time, the minimum value indicates the position of best match. Smaller values of the sum of normalized differences represent higher similarity score.

In our algorithm, since the matching primitives are 1-D horizontal intensity line segments, the similarity measure is carried out in one dimension only. The source and target segments are on the corresponding scanlines (epipolar lines), the sum of normalized differences function becomes:

$$d(s) = \sum_x \left| \frac{f(x)}{\bar{f}(x)} - \frac{w(x-s)}{\bar{w}} \right|$$

Moreover, the purpose of normalizing for intensity changes is to allow for changes in contrast or illumination when taking the left and right images separately. However, for general purpose, the illumination and contrast change between the

two images should not be large. Therefore, a threshold check can be performed to the value of $|\bar{f}(x) - \bar{w}|$, so as to disambiguate and reject candidate matches that are close to the source in shape of waveform but with very different intensity or contrast. This does not require much additional computations because $\bar{w}$ and $\bar{f}(x)$ are already calculated in the original sum of normalized differences function. This similarity measure is used in the implementation of the algorithm.

When comparing a source segment with one of the candidate segments, we do not need to slide the two segments over each other as in the above examples (refer section 5.1). The similarity measure is performed only once between the source segment and each of its candidate segments. We can assume that the two input segments are of the same length and the similarity measure is implemented to perform on two segments of the same length. In the matching scheme to be presented in the rest of this chapter, segments of different length are transformed to the same length before performing the similarity measure. The following sections explain the reasons and describe the detail operations. Finally, every candidate segment finally get a single similarity score with the source segment. The candidate segment with the best score is made the match (correspondence) of the source segment.

5.3 Treating Inclined Surfaces

A horizontal intensity line segment corresponds to part of a surface in the real 3-D environment. For planar surfaces parallel to the two imaging planes , the surfaces should project into the two imaging planes as similar surfaces of the same size, with certain displacements. This is also true for surfaces inclined only in the vertical direction. This is depicted in fig. 5.1. If a 3-D surface is not inclined in the horizontal direction, its left and right sides should produce the same disparity in the left and right images. The horizontal intensity line

segments extracted from such surfaces thus have equal length in the left and right images.

Corresponding pair of segments with different length results from 3-D surfaces inclined in the horizontal direction (fig. 5.1). If the 3-D surface is inclined in the horizontal direction, the left and right side of the surface produce different disparities in the two images. However, the difference in disparities should not be large. i.e., the difference in length in the corresponding line segments extracted from such surfaces in the left and right images should not be large. Therefore, before comparing two segments using the similarity measure function, we can first check the lengths of the segments. If the difference is large, we can reject the match and reduce the search space. Note that this is actually a constraint on the shape of the objects in view similar to the ordering constraint mentioned in section 1.4.5. It facilitates the assumption of simple surfaces instead of more complicated ones in case of ambiguity.

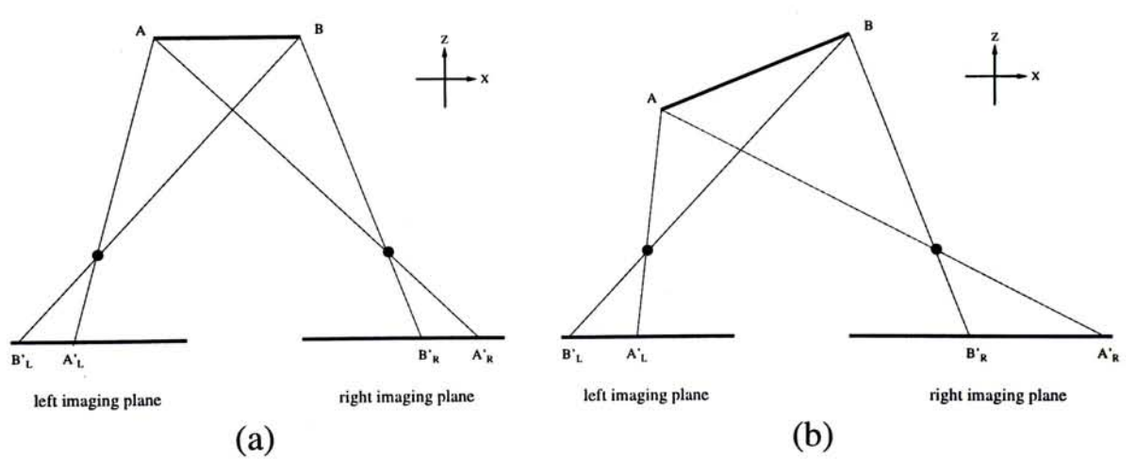


Figure 5.1: Projection of 3-D world planar surfaces to imaging planes; (a) surface not inclined in horizontal direction; (b) surface inclined in horizontal direction. Note that in (b), the projection of the surface on the two imaging planes are of different length.

When a source segment is matched with one of its candidate segments and a difference in length is detected (in this case the candidate segment should have

passed the threshold test), the assignment of disparity values to the pixel points along the segment should reveal the fact that the surface is an inclined one. The assignment of disparity value will be described in section 5.6. However, we also need to take into account the situation described in section 5.4.

5.4 Ambiguity Caused By Occlusion

Occlusion is one of the main problems in stereo vision systems. Occlusion occurs when part of the 3-D scene is visible in one of the imaging plane but not in the other. For those points being occluded, we cannot find their correspondences and hence their disparities. Moreover, occlusions are not readily detected and their presence confuses the matching of other points.

Using horizontal intensity line segments as matching primitives, occlusions may result in the following cases:

1. A line segment being visible in one image, but not in the other.

2. A line segment being visible in one image, but only part of it is visible in the other.

In the first case, there is no way of finding the disparity of the line segment, and hence there is no way of finding the disparities of the points along the segment. These segments should be detected and their disparities are not used in building the disparity map. Section 5.9 addresses more on this.

The possibility of the second case results in ambiguity. Figure 5.2 gives an illustration of two situations. Fig. 5.2 (a) shows two linked surfaces, there is no occlusion between them. Fig. 5.2 (b) shows two surfaces, one being occluded by another. In both cases, one surface produces projections of different length in the left and right images. Since both cases result in projections (segments) of different length in the two imaging planes, we need a procedure to disambiguate

them. Consequently, a scheme of comparing segments of different length is investigated and presented in section 5.5. The assignment of disparity values to the pixel points along the line segments can then be performed appropriately according to the result of the disambiguating process (the matching scheme).

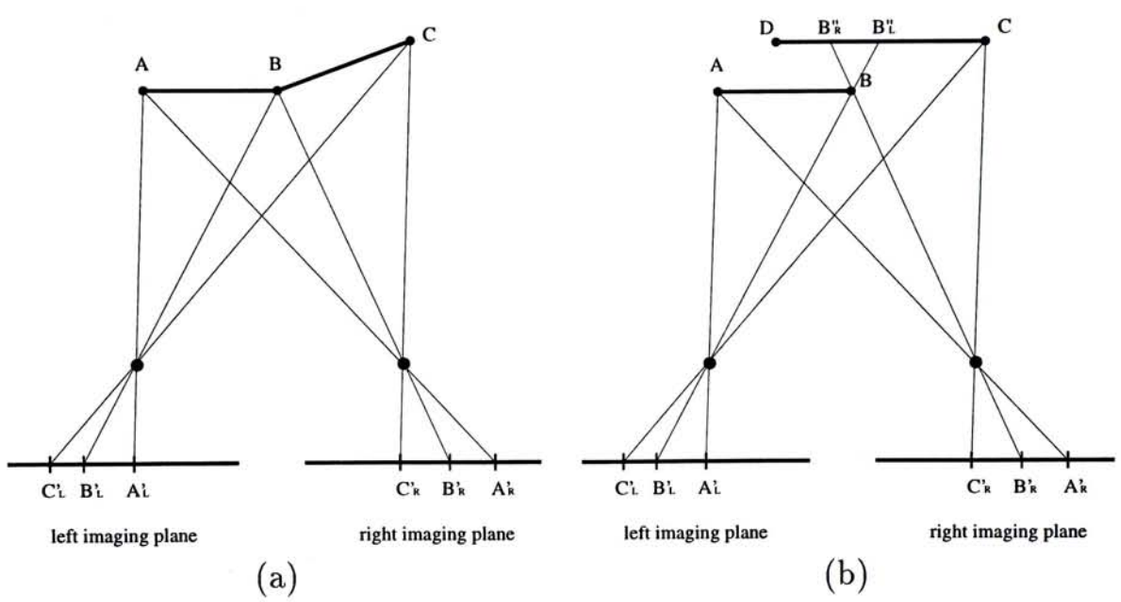


Figure 5.2: Ambiguity caused by partial occlusion: (a) two linked surfaces (one inclined); (b) two surfaces, one being occluded by another. In both cases, one surface produces projections of different length in the left and right images (B'C').

It should be noted that the situation may be more complex than that illustrated in fig. 5.2. However, in order to simplify the problem and to allow efficient disambiguation of the ambiguity caused by partial occlusion, it is assumed that either one of the cases as illustrated in fig.5.2 occurs. The matching result represents an approximation to the real case. Fig.5.3 shows a more general case.

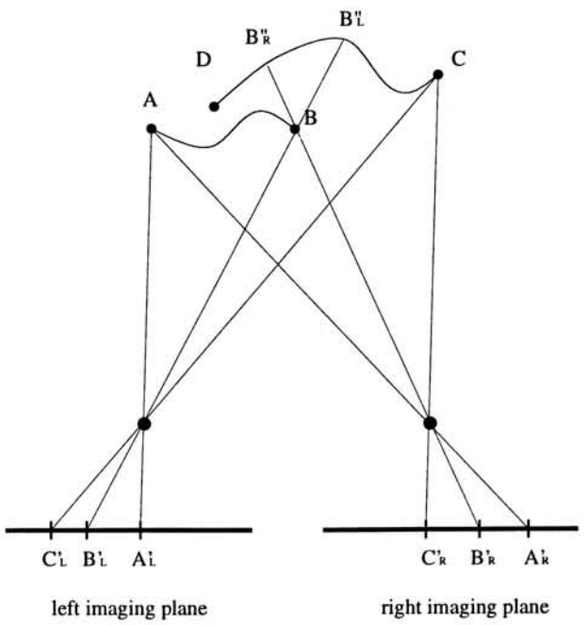


Figure 5.3: A more general case of partial occlusion.

5.5 Matching Segments of Different Length

For the candidate segments that pass the threshold test, they are compared with the source segment one by one using the similarity measure described in section 5.2. The simpler case in which there is no partial occlusion is presented first:

5.5.1 Cases Without Partial Occlusion

We do not need to slide the shorter segment over the longer one to find the point of best match. This is because we have made use of the information of edge positions when extracting the segments. The start and end of the segments correspond to positions of edges. When we compare two segments, we do not need to slide them over each other because there should be no sharp edge within the segments themselves, the start and end of segments should not be matched to the middle of each other. Instead, the source segment is matched against

each of the candidate segments once to produce a single similarity score for each candidate segments. The candidate segment with the best similarity score is made the match (correspondence) of the source segment.

To handle the difference in length of the source and target segments, the shorter segment is resampled to the length of the longer one (fig. 5.4). Then the similarity measure can be performed between two segments of the same length. The resampling is done as linear interpolation between adjacent pixels. The matching is done this way because :

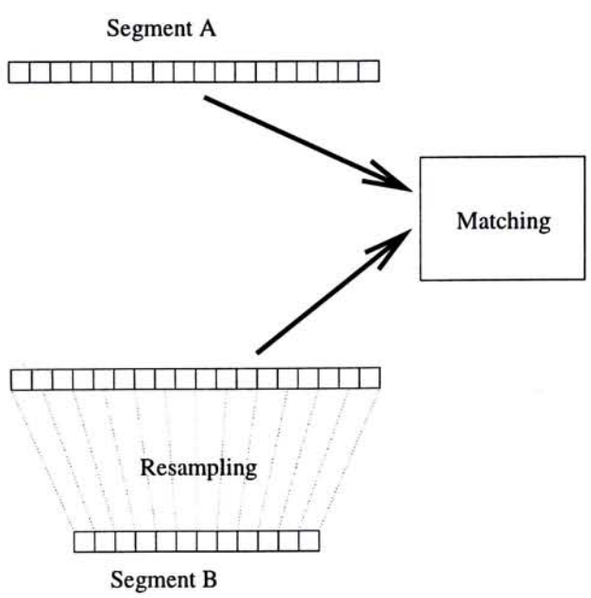


Figure 5.4: Shorter segments are resampled before comparing with longer segments

1. If two segments match each other, their end-points should match those of each other. Matching them this way is more reasonable than sliding them over each other or truncating the longer segment to the length of the shorter one.

2. The shorter segment is resampled instead of the longer one to avoid aliasing.

3. Matching the segments this way produce different disparities at the two end-points of a segment. This provides a mean to generate different disparities for the pixels along the segment, as in the case of a surface inclined in the horizontal direction.

Resampling is done as linear interpolation between adjacent pixels. This is just an approximation. In the real 3-D world, an inclined planar surface projects to the 2-D imaging planes in perspective instead of linearly lengthened or shortened. Considering a line segment resulted from an inclined 3-D planar surface, the portion of the segment that is further away from the imaging plane produce a shorter projection than the portion closer to the imaging plane. However, in the stereo vision problem, the depth of the 3-D points is the objective and we do not have a prior knowledge of the difference in depth of different parts of a line segment. The situation is even more complicated when the line segment corresponds to curved surfaces in the 3-D world. Therefore, although linear resampling is not very accurate, it is adopted as an approximation in the implementation of this matching method. The assignment of different disparity values along the line segments is described in section 5.6.

5.5.2 Cases With Partial Occlusion

However, there are cases in which truncating the longer segment to the length of the shorter one is more reasonable. These happen in cases of partial occlusion described in section 5.4. In such cases, one end (either left or right) of the two segments match each other, but not the other end (see fig. 5.2). In these cases, only part of the longer segments can be matched to the shorter ones, the other part are being occluded in the other image, thus causing the difference in length. To deal with these segments, the two segments to be compared are aligned on the left or right end, and the longer one is truncated to the length of

the shorter one. The shorter segment is not resampled as in the cases with no partial occlusion.

Since we do not have a prior knowledge of whether the segments match on the left or right, we need to compute the similarity measure twice. First, the segments are aligned on the left and the longer segment truncated to the length of the shorter one. Then, the process is repeated with the segments aligned on the right. Each of the two runs of the similarity measure simulates the case of partial occlusion on the right end and on the left end of the longer segment respectively. The similarity measure among the two that give a higher similarity score indicate a better point of match of the two segments, and hence a more likely-to-be-correct hypothesis of alignment and partial occlusion. Therefore, the two case (left and right alignment) is disambiguated using the similarity score.

5.5.3 Matching Scheme To Handle All the Cases

A matching scheme is used to disambiguate the above-mentioned cases, and at the same time find the matches (correspondences) of the points in the images. When we compare between two segments of different length (a source segment and a target segment, one among a number of candidate segments of the source segment), we compute the similarity measure three times.

First, we have the shorter segment resampled to the same length as that of the longer segment, the longer segment and the resampled segment are passed to the similarity measure and we have a similarity score S_1. Then, the two segments are aligned on the left and the longer segment is truncated on the right to the length of the shorter segment, the resulting segments are passed to the similarity measure and we have another similarity score S_2. Finally, we have the segments aligned on the right and have the longer segment truncated on the left. The resulting segments are passed to the similarity measure and we have

similarity score S_3. The similarity scores should be normalized for the length of the segments, because otherwise the result will be biased because the matching length for S_1 is that of the longer segment and those for S_2 and S_3 are that of the shorter one.

S_1 corresponds to the case where there is no partial occlusion. For convenience, we call it *inclined* type matching. S_2 corresponds to the case where there is partial occlusion on the right, we call it *left* type matching because the segments are aligned on the left. Similarly, S_3 is called *right* type matching, it corresponds to the case of partial occlusion on the left.

The best of the three similarity scores will be the final similarity score of the particular candidate segment. The candidate segments then compete to be the match of the source segment. If a particular candidate segment is finally the match of the source segment (i.e. having the best similarity score among all the candidate segments), assignment of disparity values to the pixel points along the segment will be carried out according to the fact that whether the final similarity score is of the type *inclined* , *left* or *right* .

5.5.4 Matching Scheme for Segments of same length

If the source segment is of the same length as that of the target segment, the similarity measure need only be computed once. The segments are aligned on both end and no sliding of segments is necessary. This is already handled in the above-mentioned matching scheme because segments of the same length will result in equal similarity scores no matter the type of matching is *inclined* , *left* or *right* . Of course, the similarity measure need not be computed three times in implementation.

With the matching scheme capable of handling all kind of segments mentioned above, all the source segments are matched one by one with their candidate segments and then the disparity map can be built.

5.6 Assigning Disparity Values

When correspondence is established between two points on the left and right images, disparity of the corresponding world point can be found by simply measuring the displacement between the corresponding points in the two images. With the parallel-axis imaging geometry described in section 1.4.3 and assuming epipolar constraint, the displacement occurs in the x-coordinates. Therefore, the disparity value of a world point can be found by measuring the difference in x-coordinates between its projections in the left and right images.

In the matching algorithm presented in this thesis, horizontal intensity line segments are used as the matching primitives instead of pixel points. Correspondences are found between the line segments instead of individual pixel points. After the correspondences between the segments are established, disparity values are assigned to the pixel points that compose the segments.

The scheme of assigning disparity values to the pixel points along the segments is that :

1. Disparity values of the two end-points of a segment is found. They are found by measuring the displacements in x-coordinates between corresponding segments. The left end-points and the right end-points are measured separately.

2. If the type of matching is *inclined* , disparity values of the pixel points composing the segment is found by linear interpolation between disparity values of the end-points so that disparities change gradually along the line segment.

3. If the type of matching is *left* , disparity values of all the pixel points along the segment follows the disparity of the left end-point, the unmatched portion of the longer segment is not assigned with any disparity value, i.e. they are considered to be occluded.

4. If the type of matching is *right* , disparity values of all the pixel points along the segment follows the disparity of the right end-point, the unmatched portion of the longer segment is not assigned with any disparity value.

Suppose a segment is matched with another segment in the other image and the segments have the same length, the whole line segment will have the same disparity value, no matter it is considered to be any of the three matching type. This is because the end-points of the segment will have the same displacement in the two images and every point within the line segment will get the same disparity value as that of the end-points.

In the case of a segment matched with another in the other image and they have different length, the end-points will have different displacements, and thus different disparities. In case of *inclined* type of matching, linear interpolation causes the disparity values of the pixel points to change from that of one end-point to that of the other end-point gradually and linearly. This corresponds to the real case of an inclined surface in the real world. In case of *left* or *right* type of matching, interpolation is not needed and all pixels points along the segments follows the disparity of the left and right end-point respectively. This corresponds to the real case of partially occluded surfaces on the right and left respectively. The unmatched portion of the longer segment is not assigned with any disparity value because they are the part occluded and correspondence cannot be found.

Again, this scheme has its limitations in the cases of general curved surfaces in the real world (fig. 5.3). In such cases, the surfaces are approximated by planar surfaces.

5.7 Another Case of Partial Occlusion Not Handled

Fig. 5.5 shows another case of partial occlusion not yet handled in the above-mentioned matching scheme. Partial occlusion of the same line segment occurs at both ends. The end-points of the world 3-D surface may or may not project in any of the two imaging planes. In this case of partial occlusion, it is no longer reasonable to assume any alignment of the end-points of segments. To find out the best positions of correspondence, the segments need to be slided over each other until there is a best match between the overlapping portions of the segments. In this way, the advantages of using horizontal intensity line segments as matching primitives are no longer valid. Moreover, the search space is large and the search is prone to error and it is difficult to get good results. Therefore, this kind of partial occlusion is not handled in this matching scheme.

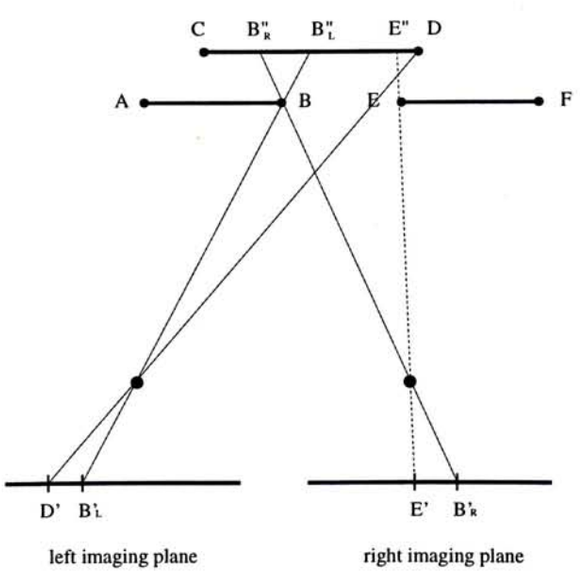


Figure 5.5: Another case of partial occlusion : partial occlusions occur at both ends.

In case of the presence of such kind of partial occlusion and there are texture

on the surface being occluded, hopefully edges can be found within the surface in some positions that can be projected onto both of the two imaging planes. Then the situation reduces to the simpler cases of partial occlusion described in section 5.4. This is because the presence of edge points provide positions for segments to align while matching. In such cases, the matching scheme still works.

5.8 Matching in Two passes

5.8.1 Problems encountered in the First pass

So far, the matching scheme described above works on the assumption that when there is a segment on the source image, we can find its matching segment on the target image unless there is occlusion. In case of occlusion, the source segment is being occluded by other objects in the target image and the corresponding segment cannot be found. However, apart from the problem of occlusion, there is another problem that can make the search for target segments fail.

The problem comes from the preprocessing stage. Remember that the preprocessing stage consists of edge detection and the extraction of horizontal intensity line segments. The images are first passed to an edge detector, then segment extraction is done according to the edge map. However, the two edge maps does not always match each other, i.e. there are some edges that are present in one image and not in the other. Since the extraction of segments is done according to the positions of edges, discrepancy in the edge maps results in discrepancy in the segments extracted. When there are edge points that are present only in one image, the segment extracted according to these dangling edge points will not have a corresponding segment in the other image.

Figure 5.6 depicts the situation. C is a weak disparity discontinuity, it can only be detected as edge C' on the left image, and is not present in the right

image. This results in the extraction of segment $A'C'$ and $B'C'$ in the left image, but just $A'B'$ in the right image. The discrepancy in edge detection may be caused by difference in illumination, contrast and weak edges that are boundary cases themselves in edge detection.

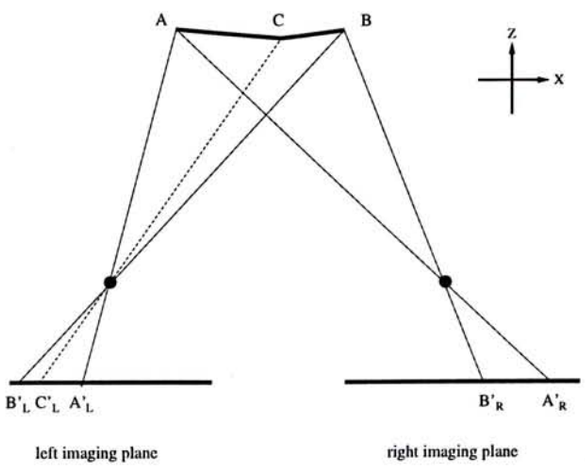


Figure 5.6: Problem arisen from edge detection. (see text)

In our matching algorithm, a source segment is supposed to match with another segment, but not two or more. Therefore, the presence of C' cause confusion in matching the segments involved. One possibility is that the segments can still be matched with other segments but the matching is a mismatch. However, a more likely possibility is that a segment find no matching segment because all of its candidate segments are out of the allowable disparity range (section 5.1). This is because this situation of dangling edges is more likely to happen within long segment. When a long segment is broken into two or more shorter segments on the other image, there is a good chance that their end-points are out of the allowable disparity range of one another's. Therefore, the situation will result in segment without being matched.

5.8.2 Second pass of matching

To handle the problem, we introduce a second pass of matching after the first pass of matching. The first pass of matching is carried out as described in the above sections in this chapter. Then a second pass is carried out to fill out the gaps resulted from the dangling edges.

In the second pass of matching, the source segments are matched one by one as in the first pass. However, the source segments are not compared with target segments or candidate segments. The source segment is compared in the target image as a sliding window. The source segment is slided within the allowable disparity range on the same epipolar line (scanline), compared with the pixels in the target image that is overlapping with the sliding source segment using the same disparity measure as in the first pass. The position where the sliding source segment gets the best similarity score is made the match of the source segment.

The second pass of matching do not take the advantage of using horizontal intensity line segments as matching primitives. More computation is required because the source segment is slided along the scanline. The similarity measure is computed once for every pixel within the allowable disparity range. There is no consideration for inclined segments and partial occlusions. The assignment of disparity values is the same to all the pixel points along the segment. Although the second pass consumes more computation per source segment than the first pass, computation is mainly spent in the first pass because the ratio of dangling edges is small compared with all the edges in the images. Therefore, the second pass of matching only act as a remedy of the problems met in the first pass. The first pass is still the more important part of the own matching algorithm.

5.9 Refinement of Disparity Map

Using the matching scheme described above, a coarse disparity map that roughly contains the disparity values of the individual pixel points can be obtained. However, further refinement of the disparity map can improve the result.

Occlusion One of the problem that give rise to the necessity of refining the disparity map is occlusion. Occlusions comes from the fact that some world points are visible in one image but not in the other. It has been discussed in section 5.4 that with our choice of using horizontal intensity line segments as the matching primitives, occlusion may results in :

1. A line segment being visible in one image, but not in the other.
2. A line segment being visible in one image, but only part of it is visible in the other.

The latter case has already been discussed in section 5.4 and has been handled by the matching scheme described in the above sections. In the former case that a line segment visible in one image is wholly invisible in the other image, there is no way of finding the disparity of the line segment, and hence there is no way of finding the disparities of the points along the segment. These segments should be detected and their disparities are not used in building the disparity map.

A way to detect such wholly occluded segments is to do a reverse matching. When a correspondence for a particular line segment on the left image is found (forward match), the matching process is performed once again using the line segment found in the right image (the best matching segment of the left source segment) as source segment and the left image as target (reverse match). If the correspondence found in the reverse match process does not agree with the result of the forward match process, it is likely that there is an occlusion and a reasonable match cannot be found. The process is depicted in fig. 5.7. This

is because segments that are wholly occluded in the other image are dangling segments with no match. When a dangling segment is matched among its set of candidate segments, there is still one candidate segment that has the greatest similarity score and that candidate segment is made the correspondence of the dangling segment in the forward match. However, actually the candidate segment should match another segment in the source image, thus a reverse match is helpful to reject those dangling segments.

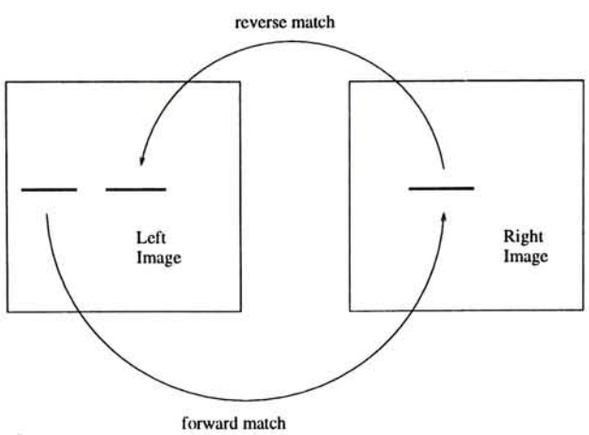


Figure 5.7: Occlusion detection process.

Disparity values at edge points Throughout the matching algorithm described so far, the edges are only used in the extraction of the line segments for matching. The edges are not matched directly and the disparity values at the edge points are not assigned. The necessity of finding out the disparity values at the edge positions is low because :

1. Only a very small proportion of all the pixels in an image are edge pixels.
2. It can be expected that the disparity range of the edge points would not exceed that of the non-edge pixels.
3. Some of the edge points are in fact results of disparity discontinuity, it is not very meaningful to fix a certain disparity value for those edges. The

disparity values of the points around those disparity discontinuity points are more important.

If it is desired to fill in the disparity values at the edge points. A separate edge-based (feature-based) stereo matching scheme can be employed to find the disparity values at the edge points. Alternatively, the disparity values at the edge points can be approximated by an average of those of its adjacent pixel points.

Smoothing the disparity map A smoothing process can be applied to the disparity map to maintain the disparity continuity constraint within area bounded by edges. This conforms our assumption that abrupt changes in disparity are unlikely in these area; otherwise edges would have been found. The vertical disparity continuity constraint can also be considered here, e.g. an area bounded by an upper and a lower edge should have similar disparity. The smoothing can be done as an averaging of the disparity values within area bounded by edges, with the discarding of extreme values.

Chapter 6

Coarse-to-fine Matching

Hierarchical structures have long been used in many stereo matching algorithms to impose global consistency in the disparity map. [DHON89] contains a review on a few of them. A hierarchical structure called the Wavelet Representation is studied and used in our matching algorithm.

6.1 The Wavelet Representation

A pyramid is a data structure that contains successively condensed information of an image. Many image representations can be viewed as pyramids. They generally contain the original images and successively lower resolution versions of the original images. Conventionally, the original image is at the lowest level, the image at one higher level is obtained by lowpass filtering and subsampling of the image at the lower level, and the process repeats for certain times. With subsampling, the image size is smaller at higher level images, hence the name pyramid. Image pyramids are useful in image processing applications including image coding and analysis. In stereo matching, they can be used to impose global consistency and speed up of the matching process.

There can be many different image pyramids depending on what lowpass

filter is used and how the lowpass filtered image is subsampled. One traditional image pyramid is the Gaussian pyramid [BURT83]. The lowpass filter used is a 5×5 filter $h(n_1, n_2)$ given by

$$h(n_1, n_2) = h(n_1)h(n_2)$$

where

$$h(n) = \begin{cases} a, & n = 0 \\ \frac{1}{4}, & n = \pm 1 \\ \frac{1}{4} - \frac{a}{2}, & n = \pm 2 \end{cases}$$

Burt proposed to use 0.4 for a, then $h(n)$ has approximately Gaussian shape, therefore it is called Gaussian Pyramid. The subsampling is done by a factor of 2 in each dimension, i.e. by a factor of 4 in two-dimensions. The Gaussian pyramid was first used as an intermediate structure to generate the Laplacian pyramid, which is mainly used in image coding. Every level of the Laplacian pyramid is the difference of two consecutive levels of the Gaussian pyramid, except at the highest level. The highest level of the Laplacian pyramid equals the highest level of the Gaussian pyramid.

In our algorithm, a pyramidal structure called the Wavelet Representation is studied and used. The lowpass filter used come from Wavelet Transform Theory. In particular the $D4$ filter by Daubechies [RIOU91] is used. A brief introduction of the Wavelet Transform is included in Appendix A.

The decomposition of signal using the Discrete Time Wavelet Transform is a scheme of subband coding. Fig. 6.1 depicts a one-dimensional example of a subband coding scheme. With a discrete lowpass filter $h(n)$, we can construct a corresponding highpass filter $g(n)$ where $g(n) = (-1)^n h(L-1-n)$, where $0 \leq n \leq L-1$ and L is the length of the filters, which must be even. Using these two filters, we can construct 4 doubly indexed filters for a 2-dimensional function [RIOU91]:

$$h(i,j) = h(i)h(j)$$

$$g^{(1)}(i,j) = h(i)g(j)$$

$$g^{(2)}(i,j) = g(i)h(j)$$

$$g^{(3)}(i,j) = g(i)g(j)$$

If we choose $h(n)$ such that $g(n)$ and $h(n)$ are orthonormal (e.g. the $D4$ filter), the 4 filters are orthogonal to each other. This follows from the orthogonality of the one-dimensional filters. The input image is convoluted with the 4 filters and 4 images are produced. $g^{(1)}(i,j)$ corresponds to taking lowpass for row and highpass for column. This effectively gives the high frequency components of the input image at the vertical component. Similarly, $g^{(2)}(i,j)$ gives the horizontal high frequency components and $g^{(3)}(i,j)$ gives the diagonal high frequency component. $h(i,j)$ gives a blurred image of the input image since it is lowpass at both the horizontal and vertical component. The filtered images can be subsampled at all 2 directions by a factor of 2, i.e. we pick one sample from every two. There is thus an overall subsampling by a factor of 4 in the two-dimensions. In this way, the filtered image will be of one-fourth the original size, the resolution is reduced to half and the scale is doubled.

This process can be repeated by feeding the lowpassed image into the 4 filters again. Another set of 4 filtered images are then obtained and each filtered image (now seven) is a subband image of the original image and they are of no correlation with each other. This process can go on and on and subband images are produced. All the subband images together form the wavelet multi-resolution representation. The representation is called multi-resolution because resolution is reduced to half on each pass of the filtering.

Another way of doing two-dimensional filtering is implemented by doing the filtering using one-dimensional filters [LAW92]. The process is shown in fig. 6.2.

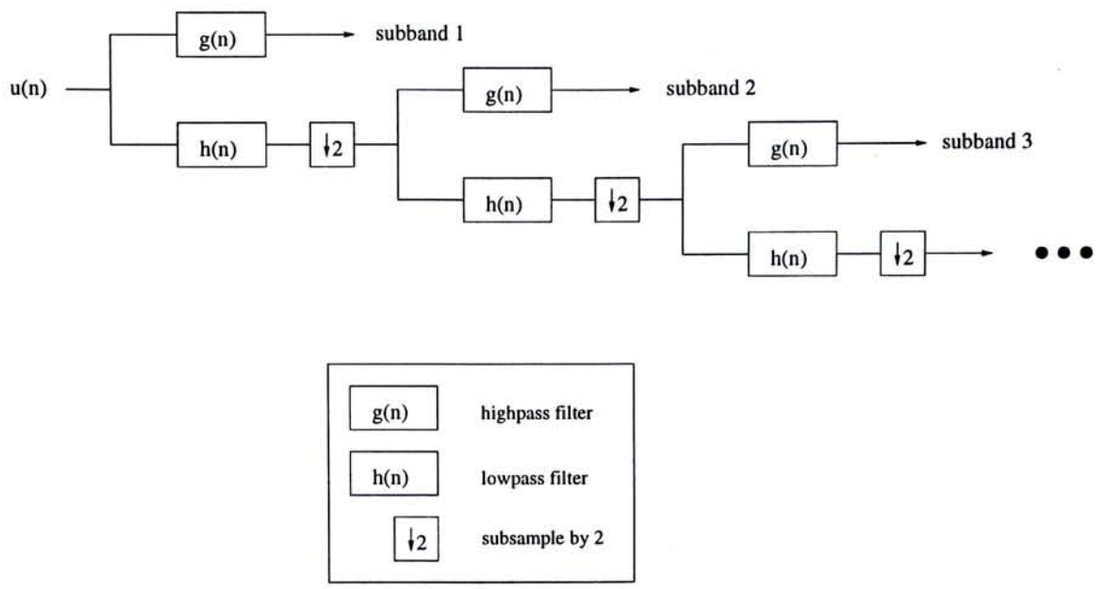


Figure 6.1: Block diagram of a subband coding scheme

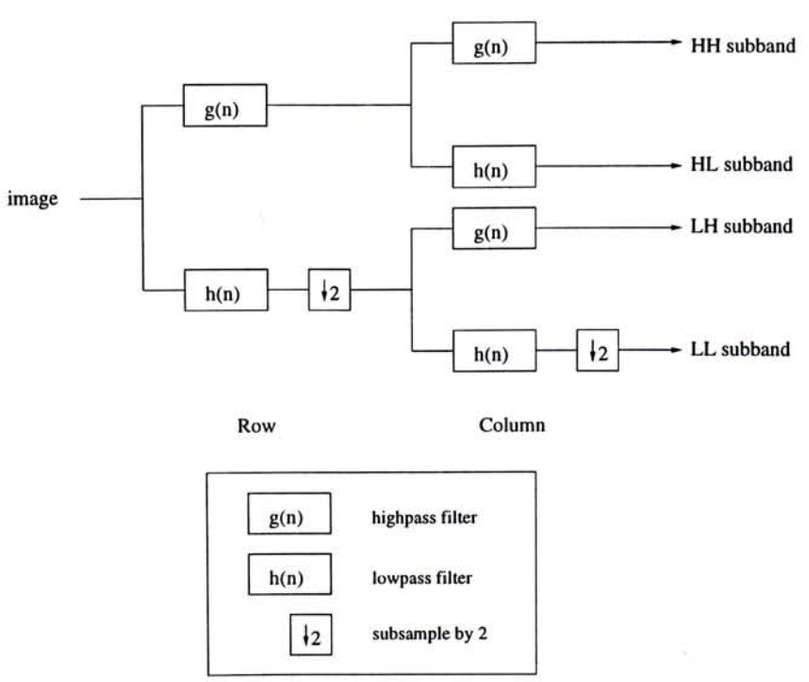


Figure 6.2: Two-dimensional filtering as 6 one-dimensional filtering processes.

Fig. 6.3 shows an original image and the images after 1, 2, and 3 passes of the process. The LL subband image refer to the result of the original image filtered by the lowpass filter in both row and column. The LH subband image refer to images filtered by the lowpass filter in row and the highpass filter in column. Similarly, the HL and the HH subband images refer to the images filtered by highpass in row and lowpass in column; and highpass on both row and column respectively. After every level of filtering, the LL image can be subsampled by a factor of 2 and then feeded to the process again to produce more subband images.

6.2 Coarse-to-fine Matching

One of the application of the Wavelet Representation is in coding. In the case of image coding, the LH, HL, HH subbands are needed to reconstruct the original image. However, in our application of using the wavelet representation in coarse to fine matching, only the LL subband images are needed. The LL images together form a pyramidal structure that the image at a higher level has the size of one-fourth that of its next lower level image. The images at the higher levels are said to be of lower resolution, on the other hand, the images at the lower levels are said to be of higher resolution. Images of lower resolution contains less details of the original images than the images of higher resolution.

With such a pyramid created using wavelet filters, a coarse-to-fine matching can thus be carried out. Starting from the top level (the smallest image), the matching is done as described in chapter 5. Then the matching result is passed to the next level (the image with size four times that of the previous level) to constraint the matching. The process goes on until the bottom level (the original image) is matched. It is called coarse-to-fine because the top level image is a coarse version of the original image, containing less details, the matching starts

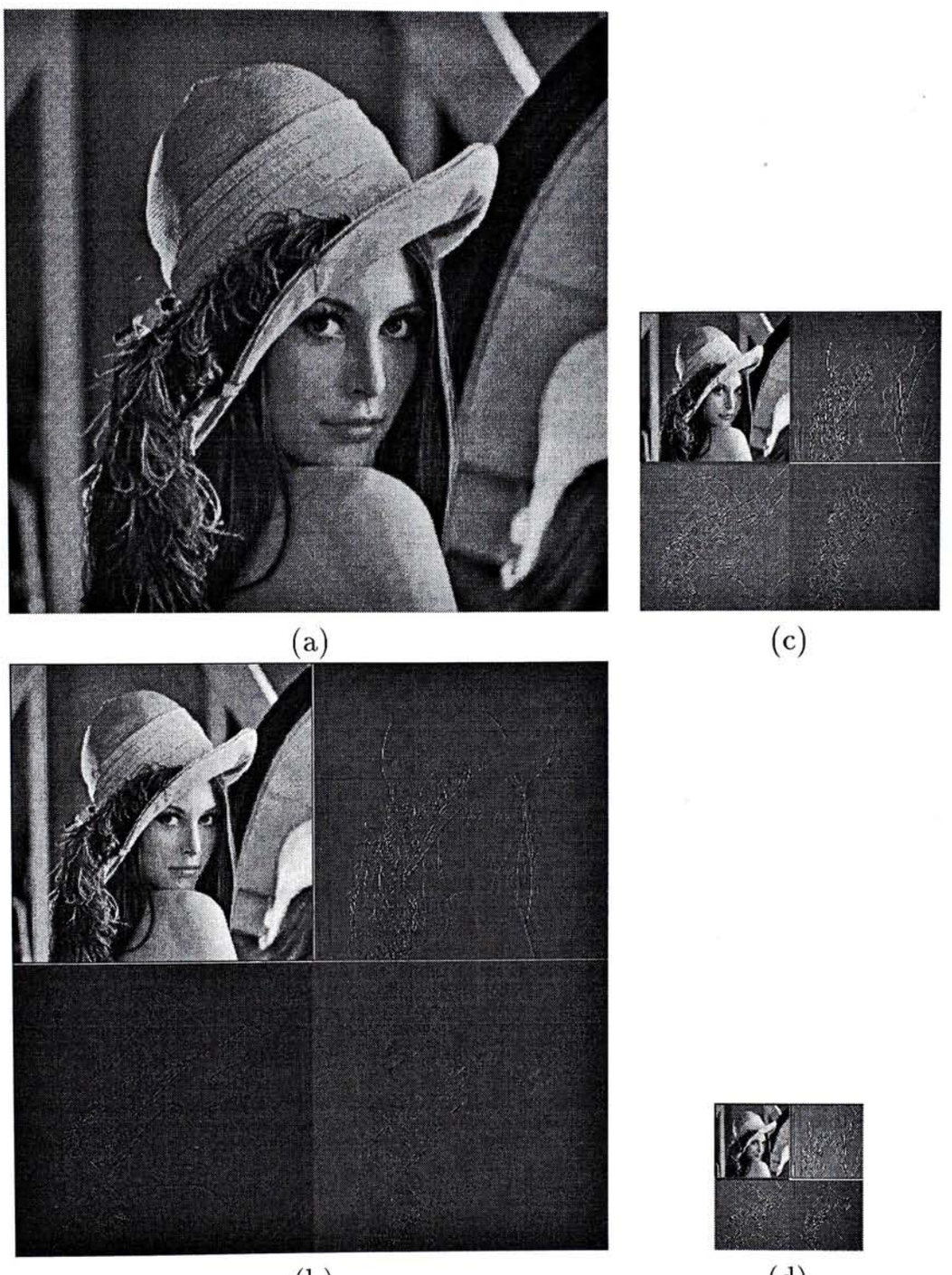


Figure 6.3: Subband images of the Lena image; (a) Original Lena image (512x512x256G); (b) Lena image after 1st pass of D4 filters (256x256x256G); (c) after 2nd pass (128x128x256G); (d) after 3rd pass (64x64x256G); (b),(c),(d) upper left - LL subband, upper right - LH, lower left - HL, lower right - HH subbands.

here and the matching result is used to constraint matching in the levels below which are of finer resolution.

The coarse-to-fine matching is done for both efficiency and accuracy considerations. First, with the constraint on lower levels from the higher levels, the matching space is lowered and the matching becomes more efficient. Second, with constraint in search space, the chance of mismatch due to similar targets within the search space is reduced. Third, with the coarse-to-fine scheme, the need for choosing a maximum allowable disparity value (section 5.1) is reduced. This is because the matching can starts at a coarse level in which the disparities are small (the image size is small), then matching at lower level are constrained by the matching at higher level, thus reducing the reliance on the setting of a maximum allowable disparity value.

On the other hand, there are possible drawbacks. First, the computation of the pyramidal structures require extra computational power, however, since the computation of the pyramidal structure is independent of the matching algorithm, speed-up such as distributed processing or pipelining is possible. Second, if a mismatch is made at a high level, the search of the correct match at lower level is hindered because of the constraint from the higher levels.

Chapter 7

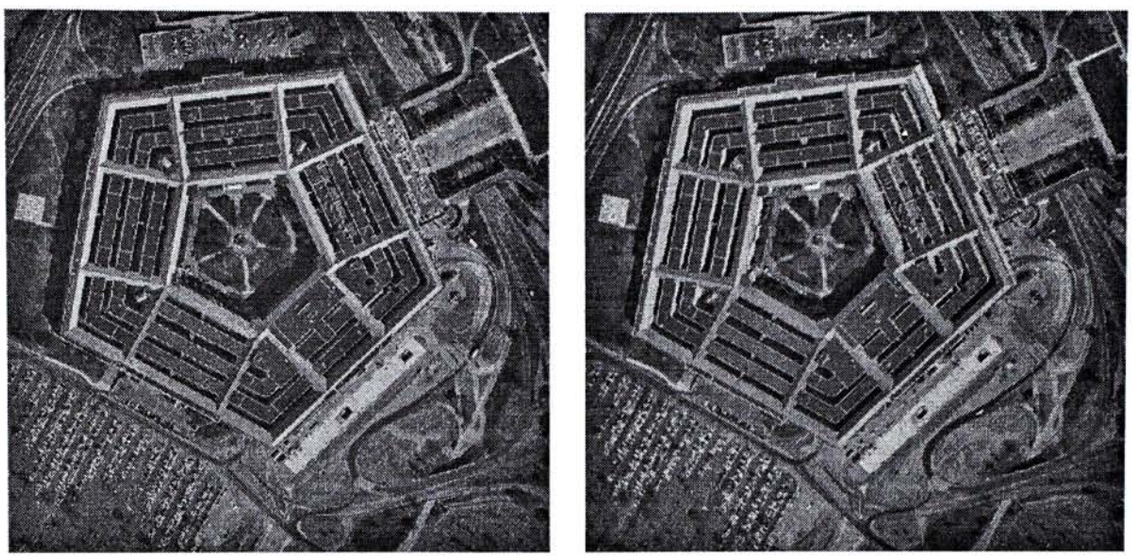
Experimental Results and Analysis

This chapter presents some experimental results of applying the matching algorithm described in the previous chapters to some testing image pairs. Some analysis of the results is also included.

7.1 Experimental Results

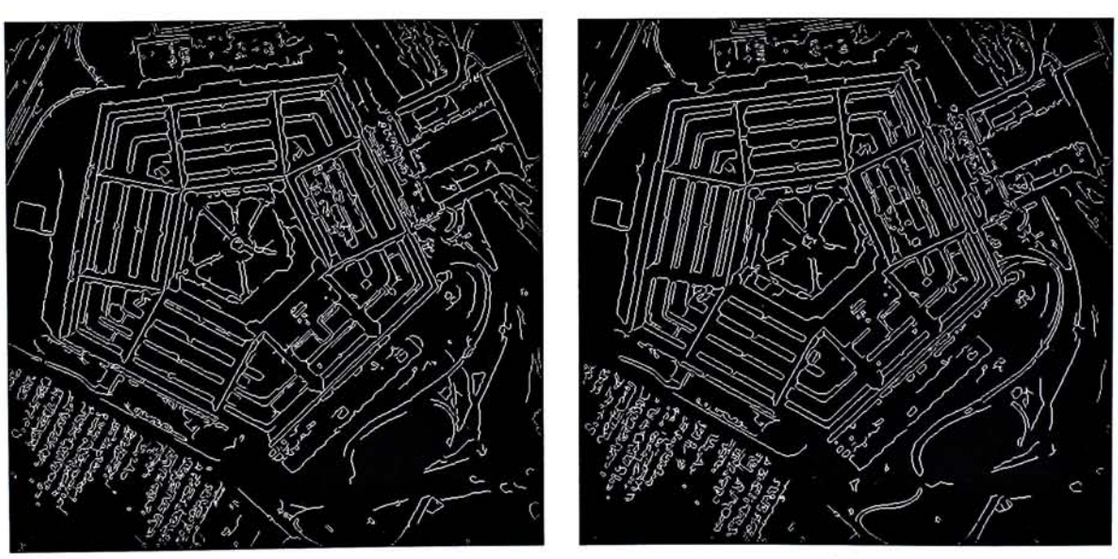
7.1.1 Image Pair 1 - The Pentagon Images

The Pentagon image pair is an aerial image pair consisting mainly of fronto-parallel surfaces. The images and data structures involved in different stages of the matching method is presented in fig. 7.1 through fig. 7.6. It shows the application of the matching method on aerial image pairs.



(a) (b)

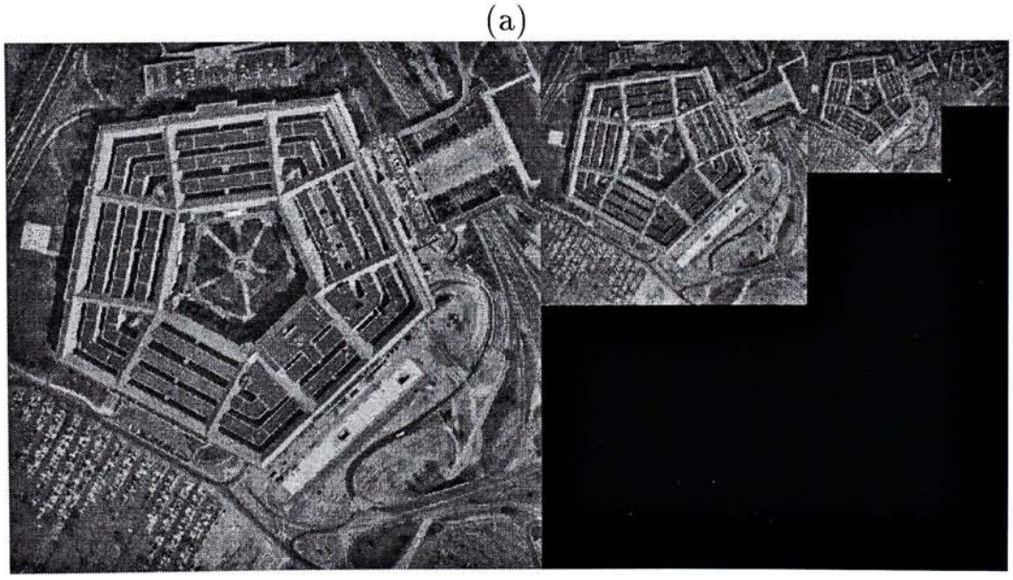
Figure 7.1: The original Pentagon Images (512x512x256G); (a) left image; (b) right image.



(a) (b)

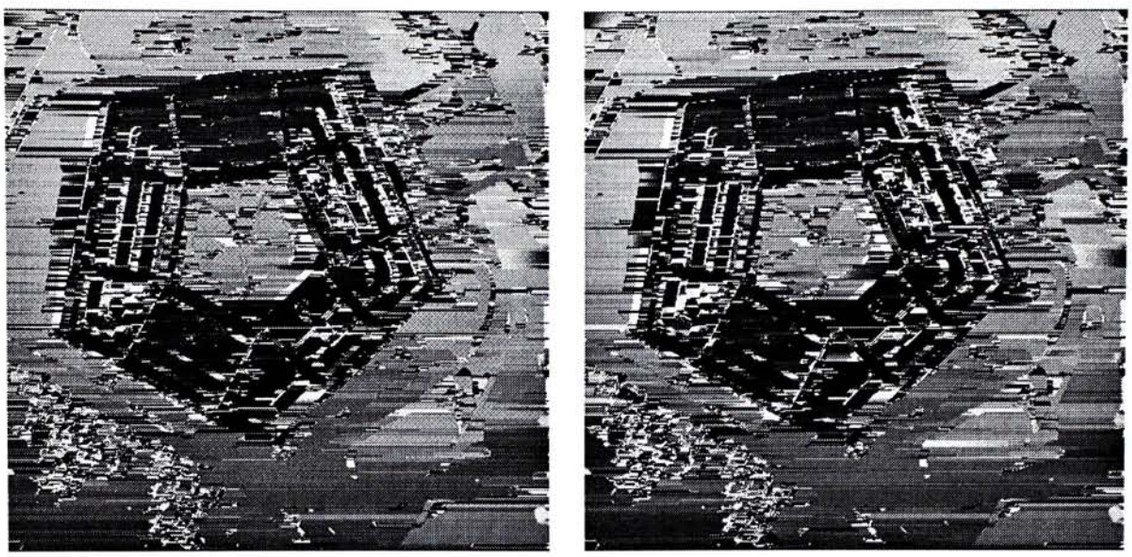
Figure 7.2: Edge images of the Pentagon images (512x512); (a) left image; (b) right image.

(a)



(b)

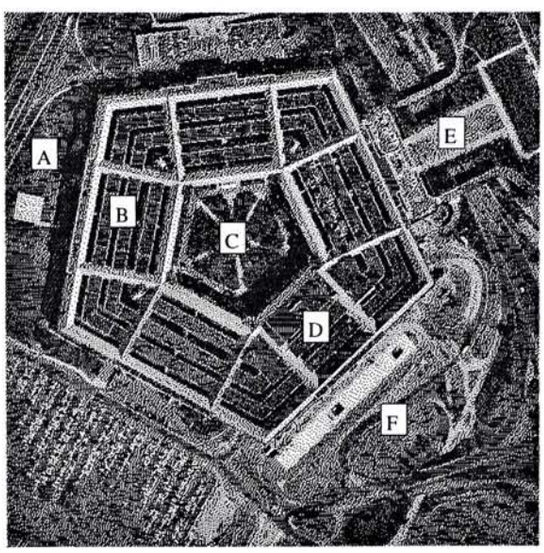
Figure 7.3: Pyramidal structures of the Pentagon images (4 levels); (a) left ; (b) right.



(a) (b)

Figure 7.4: Disparity map plotted as image, dark pixels - closer to imaging planes, light pixels - more distant to imaging planes; (a) before smoothing; (b) after smoothing.

Figure 7.5: Unmatched points due to occlusion, short segments, etc. (left)

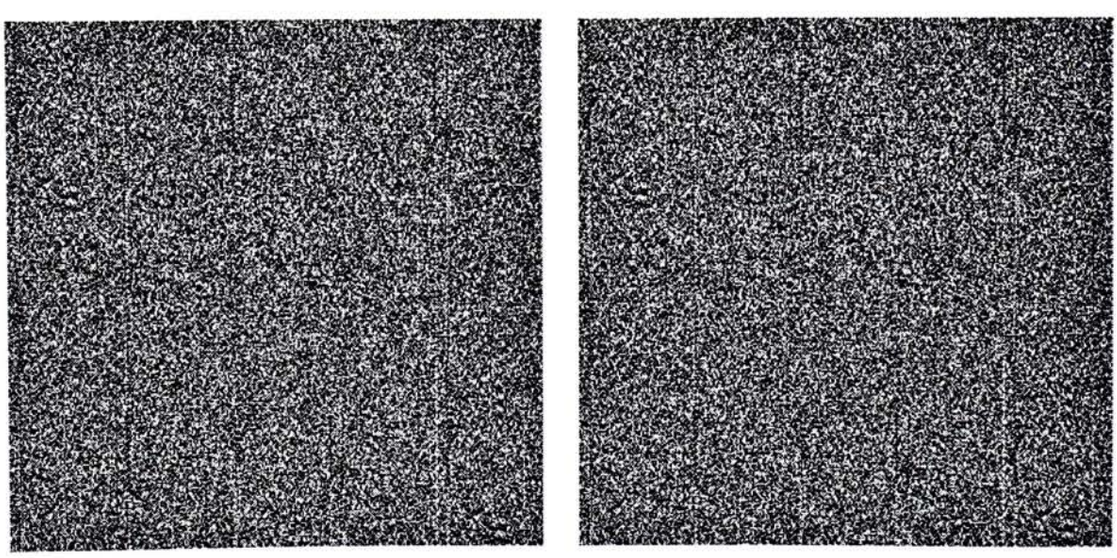


position	Disparity values given by : hand measurement	experimental result
A	+6	+6 to +7
B	-4	-4 to -3
C	+4	+3 to +4
D	-7	-7 to -6
E	+4	+3 to +4
F	+3 to +4	+3 to +4

Figure 7.6: Hand-measured disparity values and experimental results.

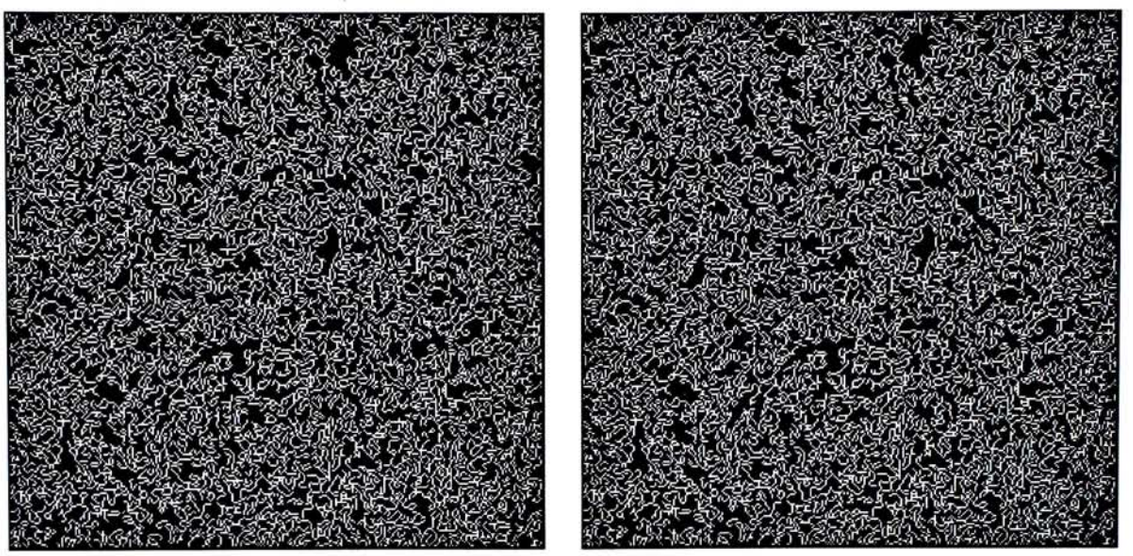
7.1.2 Image Pair 2 - Random dot stereograms

Fig. 7.7 to fig. 7.10 shows the images and data structures involved in different stages of applying the matching method on a pair of synthetic random dot stereogram. The segments are generally short in length, but the matching method still works.



(a) (b)

Figure 7.7: Random dot stereograms (320x320x256G); (a) left; (b) right



(a) (b)

Figure 7.8: Edge image of random dot stereograms; (a) left; (b) right

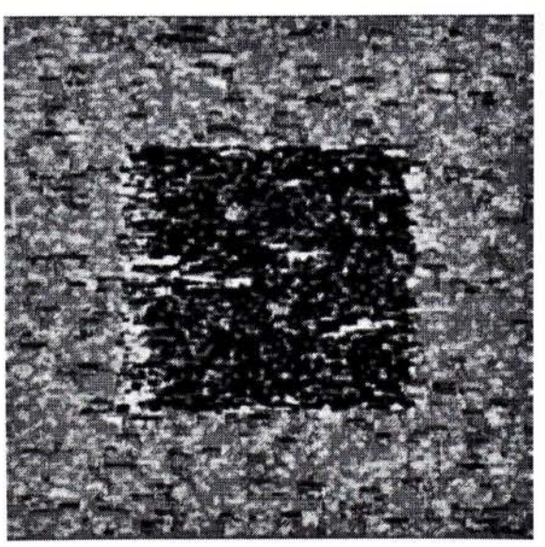


Figure 7.9: Disparity map plotted as image, dark pixels - closer to imaging planes, light pixels - more distant to imaging planes.

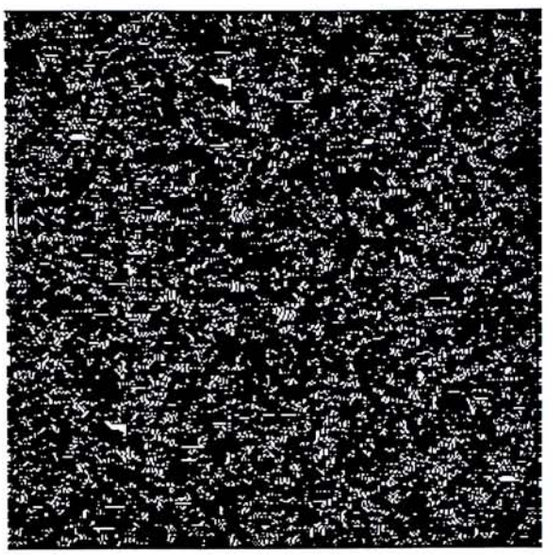
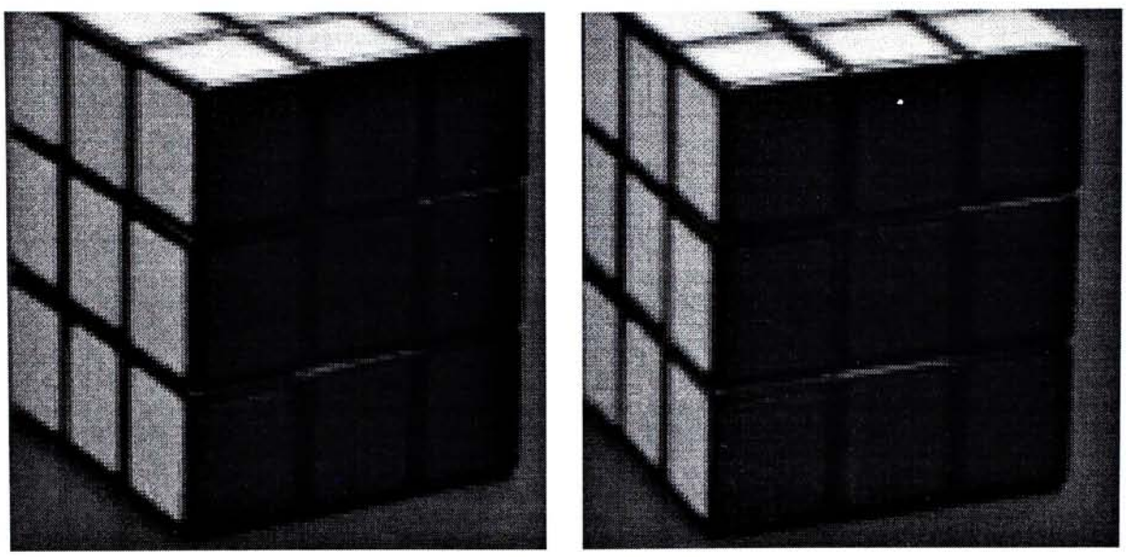


Figure 7.10: Unmatched points due to occlusion, short segments, etc. (left)

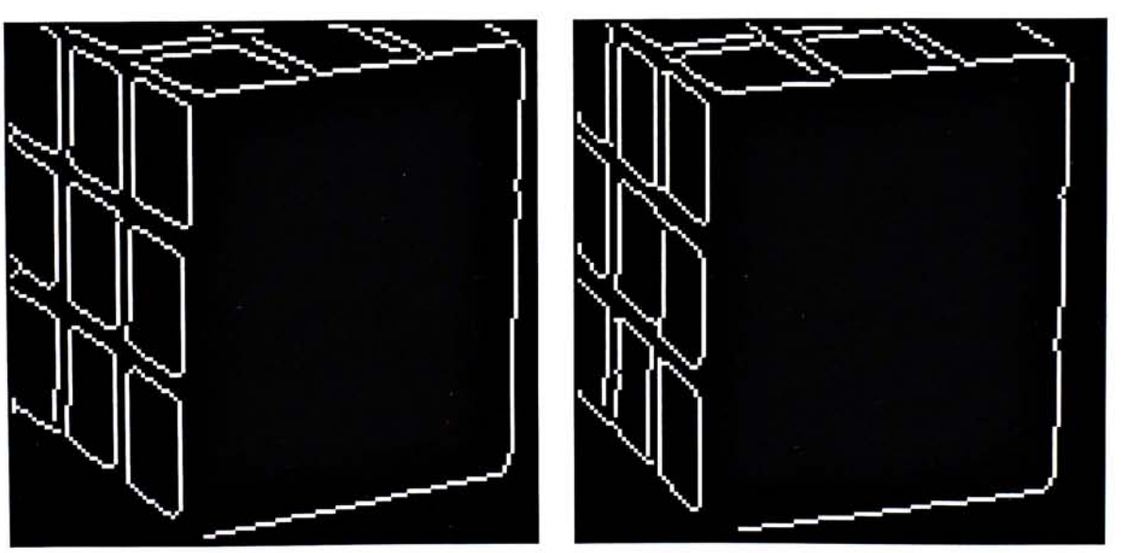
7.1.3 Image Pair 3 – The Rubik Block Images

The Rubik Block Images consist of simple block structure with inclined surfaces. This image pair is used to test the ability of the matching method in detecting inclined surfaces (fig. 7.11 to fig. 7.15). In fig. 7.13, it can be seen that the disparity values change gradually along the inclined surface.



(a) (b)

Figure 7.11: Original Rubik Block Images (132x132x256G); (a) left; (b) right



(a) (b)

Figure 7.12: Edge image of Rubik Block Images; (a) left; (b) right

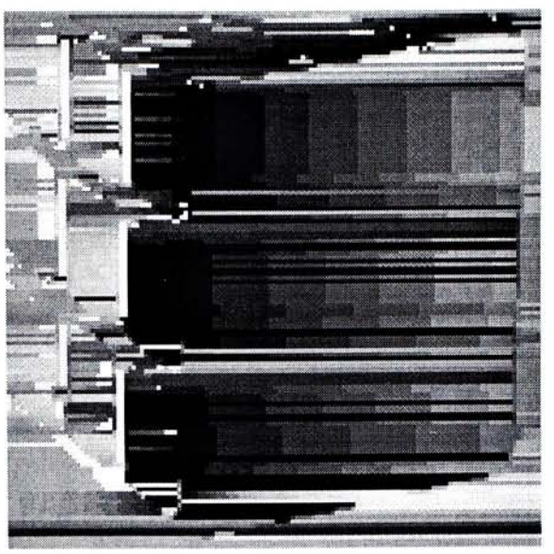


Figure 7.13: Disparity map plotted as image, dark pixels - closer to imaging planes, light pixels - more distant to imaging planes.

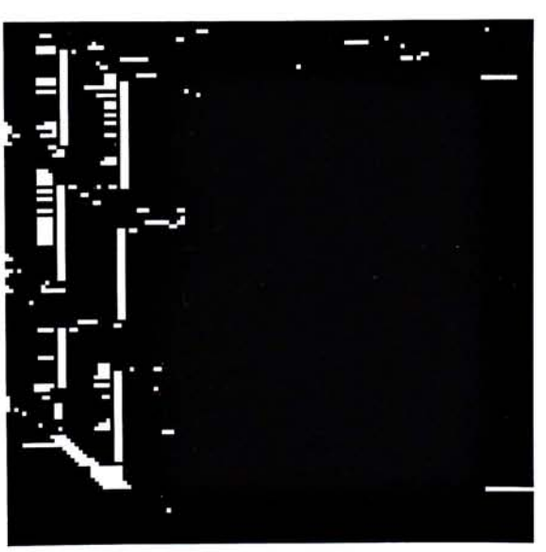
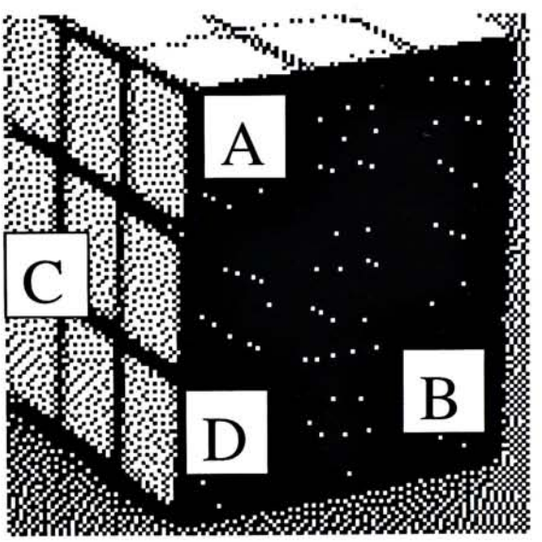


Figure 7.14: Unmatched points due to occlusion, short segments, etc. (left)



position	Disparity values given by : hand measurement	experimental result
A	-12	-12 to -11
B	-5	-6 to -5
C	-4	bad
D	-11	-11

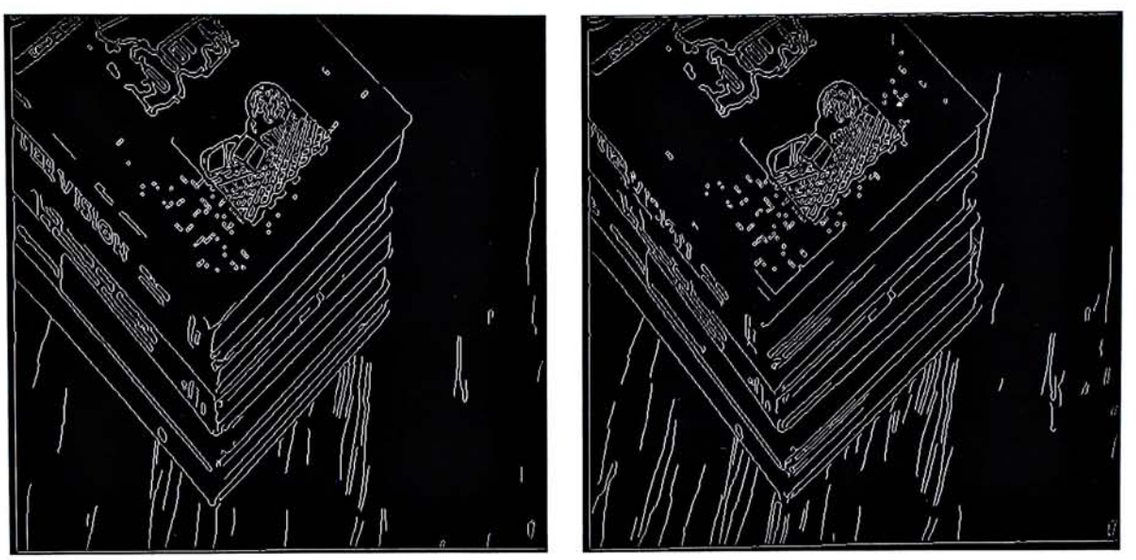
Figure 7.15: Hand-measured disparity values and experimental results.

7.1.4 Image Pair 4 - The Stack of Books Images

The Stack of Books Images contain inclined surfaces and textures that are more complicated than those in the Rubik Block Images (fig. 7.16 to fig. 7.19).

(a) (b)

Figure 7.16: Original Stack of Books Images (512x512x256G); (a) left; (b) right



(a) (b)

Figure 7.17: Edge image of Stack of Books Images; (a) left; (b) right

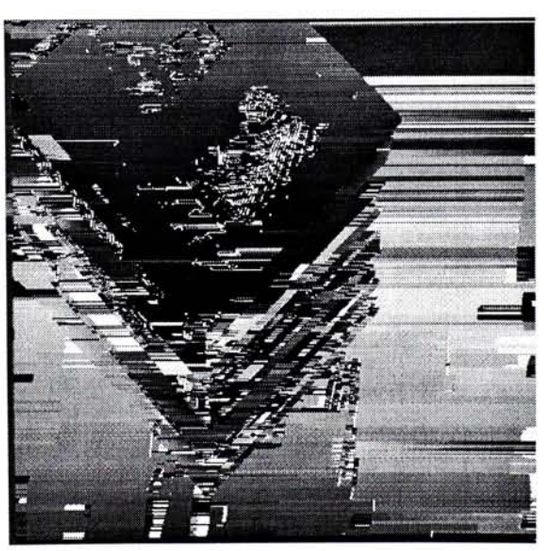


Figure 7.18: Disparity map plotted as image, dark pixels - closer to imaging planes, light pixels - more distant to imaging planes.

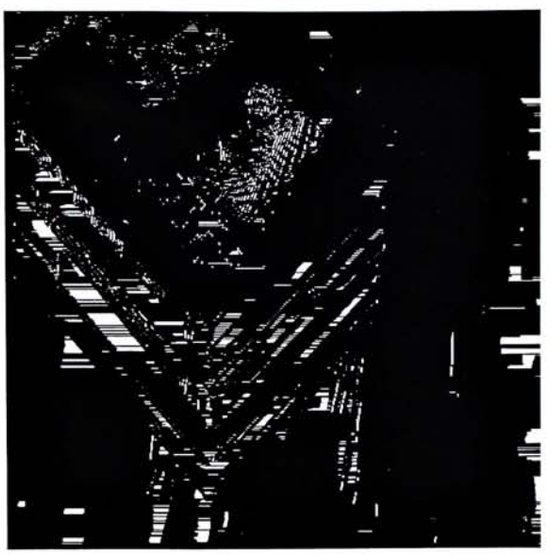
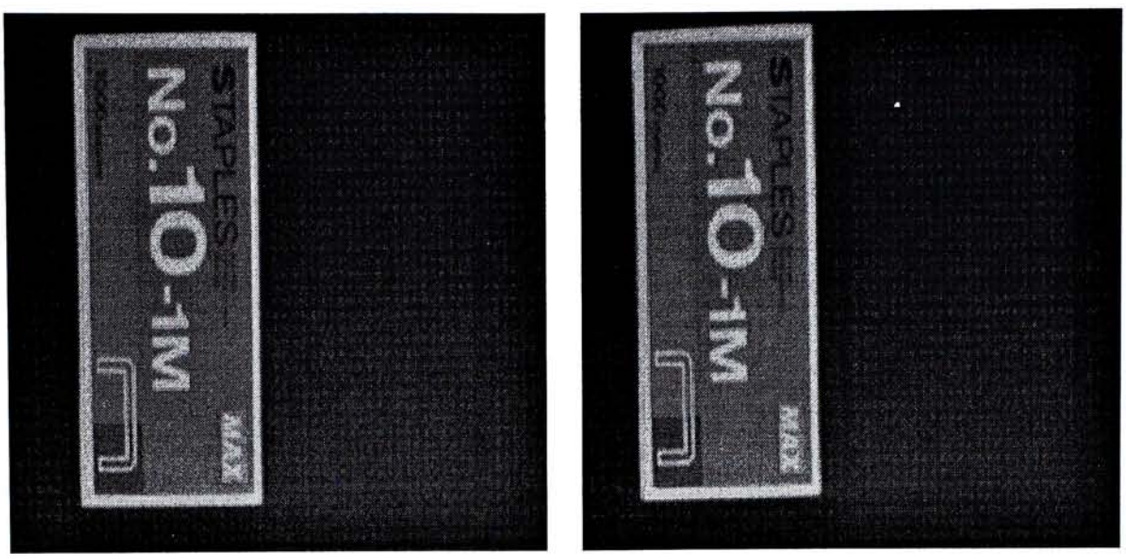


Figure 7.19: Unmatched points due to occlusion, short segments, etc. (left)

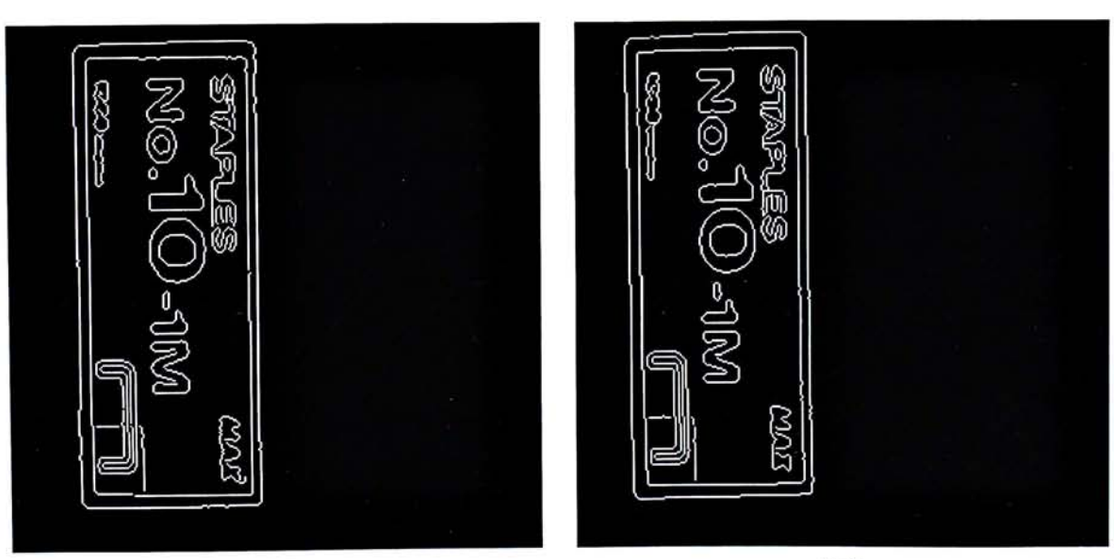
7.1.5 Image Pair 5 - The Staple Box Images

The Staple Box Images contain a fronto-parallel block structure in front of a background (fig. 7.20 to fig. 7.24). The matching method is supposed to be able to distinguish the stapler box from the background (The area left of the box should all be *left* aligned and those right of the box should be *right* aligned). As can be seen in fig. 7.23 a, it is successful in some rows. Unfortunately it fails in some other rows, resulting in error in the smoothed disparity map (the background is detected as two inclined surfaces). The large ratio of error is due to lack of texture in the background.



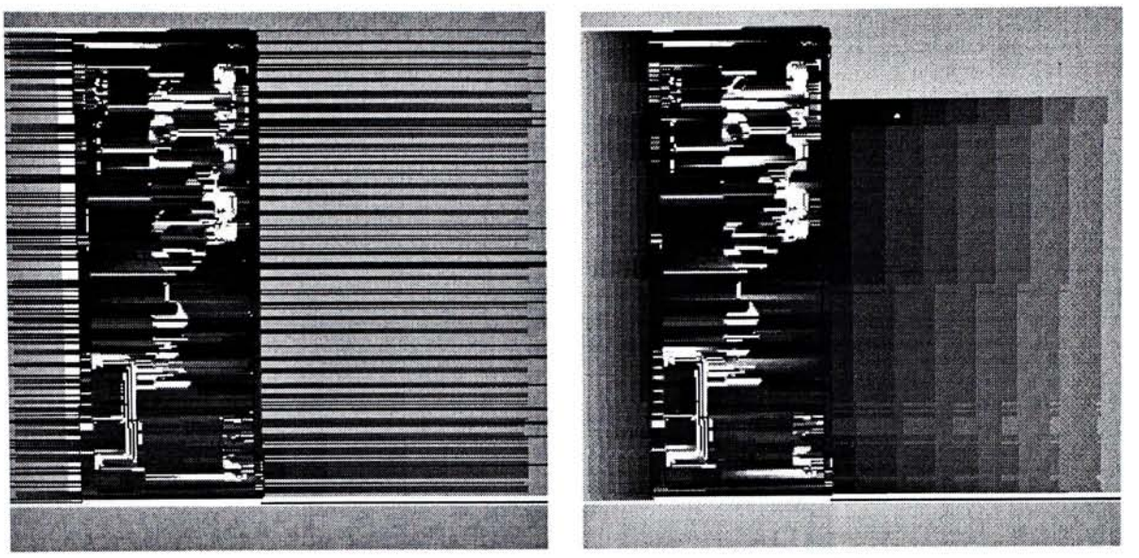
(a) (b)

Figure 7.20: Original Staple Box Images (300x300x256G); (a) left; (b) right



(a) (b)

Figure 7.21: Edge image of Staple Box Images; (a) left; (b) right



(a) (b)

Figure 7.22: Disparity map plotted as image, dark pixels - closer to imaging planes, light pixels - more distant to imaging planes; (a) before smoothing; (b) after smoothing.

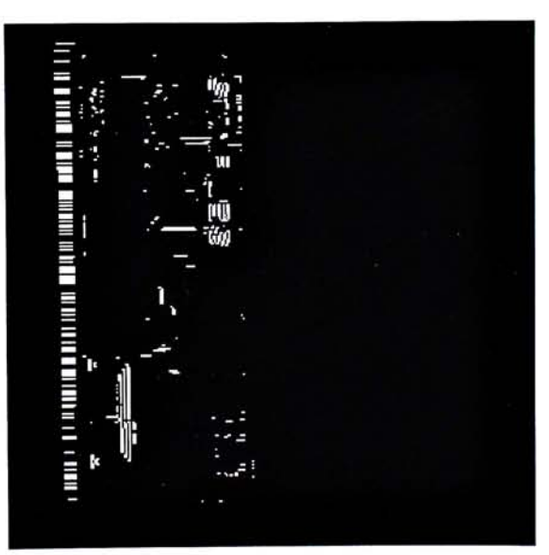
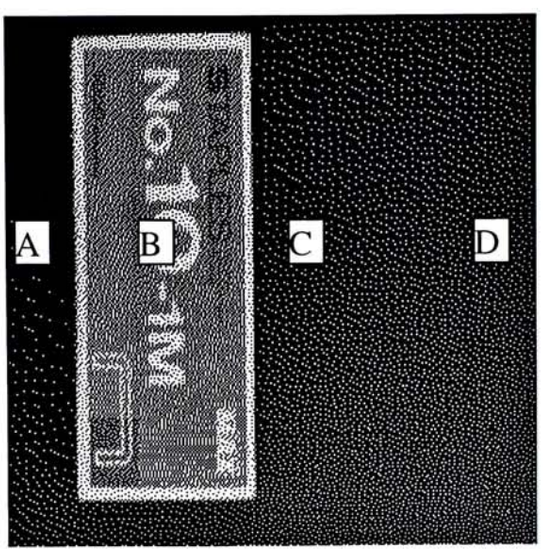


Figure 7.23: Unmatched points due to occlusion, short segments, etc. (left)

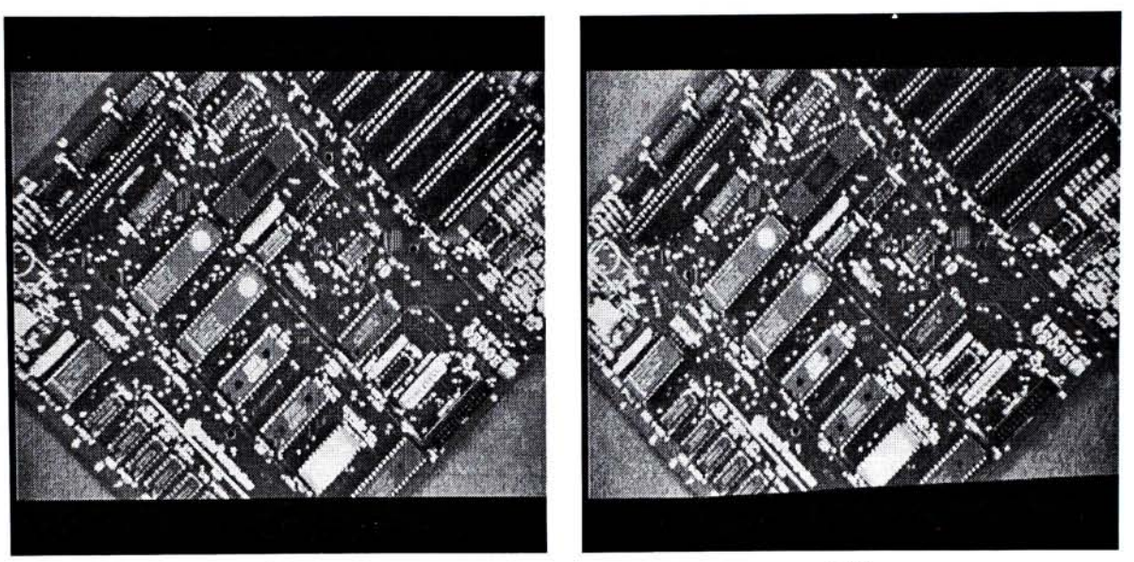


position	Disparity values given by :		
	hand measurement	experimental result	
		good	bad
A	0	0	-8 to -7
B	-8	-9 to -7	-
C	0	0	-8 to -7
D	0	0	-2 to -1

Figure 7.24: Hand-measured disparity values and experimental results.

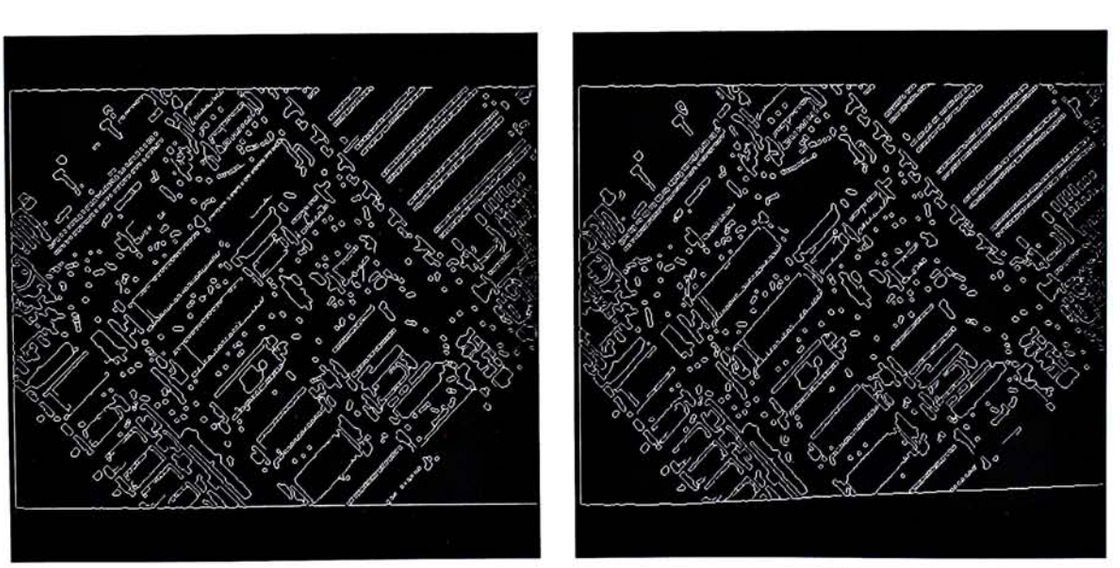
7.1.6 Image Pair 6 - Circuit Board Image

(Fig. 7.25 to fig. 7.29.)



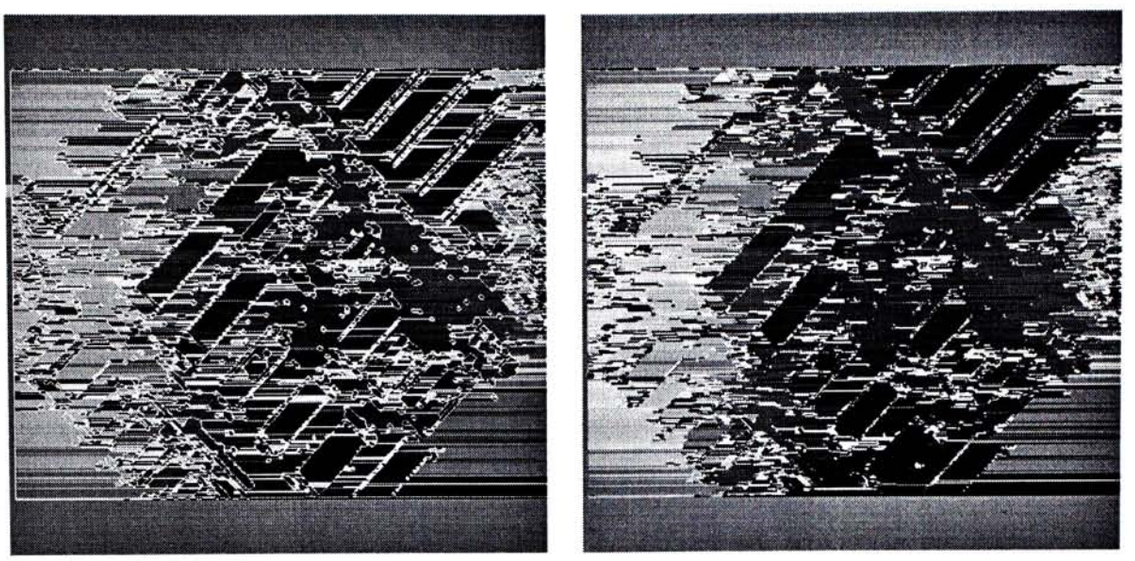
(a) (b)

Figure 7.25: Original Circuit Board Images (512x512x256G); (a) left; (b) right



(a) (b)

Figure 7.26: Edge image of Circuit Board Images; (a) left; (b) right



(a) (b)

Figure 7.27: Disparity map plotted as image, dark pixels - closer to imaging planes, light pixels - more distant to imaging planes; (a) before smoothing; (b) after smoothing.

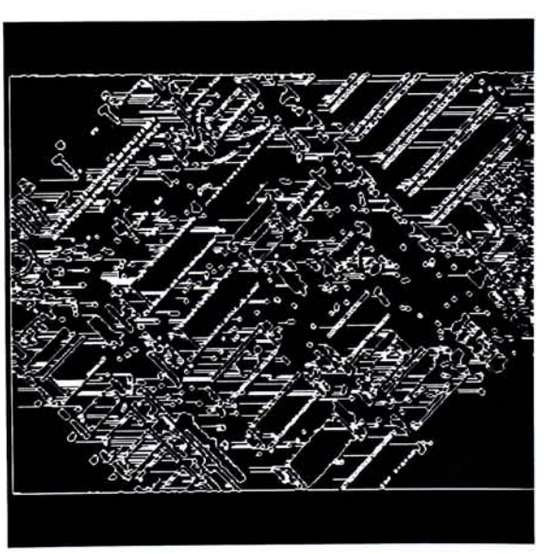
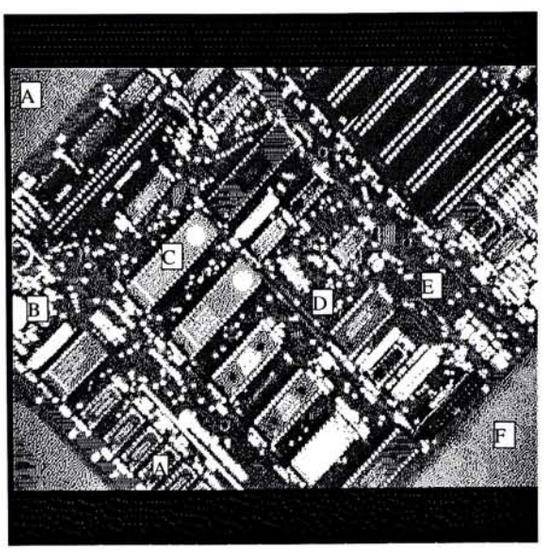


Figure 7.28: Unmatched points due to occlusion, short segments, etc. (left)



position	Disparity values given by : hand measurement	experimental result
A	0	bad
B	+11	+10 to +11
C	-3	-5 to -3
D	-5	-6 to -5
E	-3	-3 to -2
F	0	0

Figure 7.29: Hand-measured disparity values and experimental results.

Chapter 8

Conclusion

A new stereo correspondence algorithm has been studied, implemented and presented in this thesis. With consideration of the difficulties generally met by area-based stereo matching algorithms - the difficulty of finding a suitable matching window, the use of horizontal intensity line segments bounded by edge points as the matching primitives is studied. The validity of the use of such line segments as matching primitives is argued for.

Moreover, the use of such line segments contributes to a number of favorable features such as reduction of search space (hence speed-up of matching), detection and disambiguation of inclined surfaces and partial occlusions. The detection of inclined surfaces and partial occlusion is done simultaneously with the matching process rather than as a post-processing step. In particular, with analysis of the relationships between the three dimensional scene geometry and depth reconstruction, attempts have been made to give approximate results for matching stereo images of slant surfaces and partial-occluded surfaces.

The method makes use of the intensity values rather than parameterized features in the matching phase and this results in dense disparity maps. Post-processing of the disparity maps is still necessary to fine-tune the disparity values but the process is less complicated than that in feature-based methods,

in which sparse disparity maps are produced. The use of horizontal intensity line segments bounded by edges as matching primitive also get around some problems commonly encountered in correlation-based matching methods.

The method is tested on a number of synthetic images and real images. Some experimental results are presented to demonstrate the features. However, as shown in chapter 7, robustness and accuracy still need to be improved. Further study on the relationships between vertically neighboring segments and the coarse-to-fine scheme in the context of using such line segments as matching primitive may bring improvement to the results.

Appendix A

The Wavelet Transform

Transform theory has played a very important role in computer vision. Transforms are used in wide range of applications in the fields including image enhancement, restoration, encoding, compression, etc.. Among the various transforms used, the famous Fourier Transform has been a classical one. However, the Wavelet Transform has advantages over Fourier Transform in some aspects.

A.1 Fourier Transform and Wavelet Transform

For a one-dimensional signal $x(t)$, the Fourier transform :

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-2j\pi ft}dt$$

defines the notion of global frequency f in the signal. The Fourier Transform is computed as inner products of the signal with sine wave basis functions of infinite duration. Since the Fourier Transform use the sine function, which is periodic in nature, as basis function, it has difficulty with functions having transient components. If the signal has sharp transitions, the sharp transitions are spread out over the whole frequency axis in $X(f)$.

To get around the problem, the Short-Time Fourier transform :

$$STFT(\tau, f) = \int x(t)g(t-\tau)e^{-2j\pi ft}dt$$

is used. It maps the signal into a two-dimensional function in a time-frequency plane. The function $g(t)$ is a window function of limited extent, centered at time location t. With the two-dimensional time-frequency representation, the Short-Time Fourier Transform is more accurate in time when compared with the Fourier Transform. However, the accuracy depends critically on the choice of the window $g(t)$. A short window gives better time resolution but poor frequency resolution, while a long window gives poor time resolution but better frequency resolution.

Contrary to using periodic basis functions in Fourier Transform, Wavelet Transform uses basis functions that are finite in time. The basis functions are called wavelets (meaning small waves). The wavelets are obtained from a single prototype wavelet (or mother wavelet, basic wavelet) by dilation and contractions (scaling) and shifts. As a result, instead of saying that a signal is decomposed into different frequencies, the signal is said to be decomposed onto the wavelets, and since the wavelets are obtained from scaling and shifting the prototype wavelet, the signal is said to be decomposed into different scales and different locations, leading to the time-scale representation.

A.2 Continuous wavelet Transform

When a prototype wavelet $h(t)$ is used, the scaled versions of the wavelet are $h_a(t) = \frac{1}{\sqrt{|a|}}h(\frac{t}{a})$. The basis functions (wavelets) are then represented by the term $h_{a\tau}(t) = \frac{a}{\sqrt{|a|}}h(\frac{t-\tau}{a})$, where a denotes the scale and τ denotes the shift. The term $\frac{1}{\sqrt{|a|}}$ is used for energy normalization. This results in the definition of

the Continuous Wavelet Transform (CWT) :

$$CWT_x(\tau, a) = \frac{1}{\sqrt{|a|}} \int x(t) h(\frac{t-\tau}{a}) dt$$

or

$$CWT_x(\tau, a) = \int x(t) h_{a\tau}(t) dt$$

Resolution Another difference between the Fourier Transform and the Wavelet Transform is resolution. As mentioned above, a fix-sized window is used in the Fourier Transform. When the window is shorter, the time resolution is better but the frequency resolution is worse. On the other hand, when the window is longer, the time resolution is worse but the frequency resolution is better. This is known as the Uncertainty Principle. In the Wavelet Transform, this uncertainty principle still holds, however, the window size is variable with respect to frequency. This effectively give a better frequency resolution (in the expense of time resolution) at low frequency and a better time resolution (in the expense of frequency resolution) at high frequency. This is a favorable property as good frequency resolution is more significant in low frequency.

Wavelet Series Expansion In Continuous Wavelet Transform, the inverse transform is :

$$x(t) = c \int \int CWT(\tau, a) h_{a,\tau}(t) \frac{da d\tau}{a^2}$$

where c is a constant that depends only on $h(t)$. Here, both a and τ are continuous. The inverse transform is exact (i.e. the equality holds) when $h(t)$ is of finite energy and band pass. In Wavelet Series Expansion (WSE), a and τ are discrete, where $a = a_o^j$ and $\tau = k a_o^j T$ for some a_o and T. The inverse transform in this case becomes :

$$x(t) \approx c \Sigma_j \Sigma_k C_{j,k} h_{j,k}(t)$$

where c is a constant depending only on $h_{j,k}(t)$ and $C_{j,k} = \int x(t) h_{j,k}(t) dt$ is called the Wavelet coefficients. Here, the inverse transform only give approximation

of the original signal, and the accuracy depends on the quantization step of a and τ (i.e. the approximation is good when a_o and τ are small). However, it is interesting that if $h_{j,k}(t)$ are orthonormal, the reconstruction is perfect, but this cause a very limited choice of the prototype wavelets, examples are $\frac{\sin x}{x}$ and the $D4$ filter by Daubechies [RIOU91].

A.3 Discrete Time Wavelet Transform

To make Wavelet Transform useful in image processing, we need the Discrete Time Wavelet Transform ($DTWT$), where both the signal and the wavelets are discrete. The $DTWT$ is a subband coding scheme. The $DTWT$ has advantages in computation, instead of having many different wavelets as scaled versions of the prototype wavelets, we can use the same discrete valued basis function $h(n)$ with subsampling of the input signal. Here, Sub-sampling is done by taking one sample out of every two. The size of the input signal will then be reduced to half and the resolution is also reduced to half. The scale is changed by sub-sampling of the input signal instead of changing a in the continuous case.

We can easily compute another function $g(n)$ where $g(n) = (-1)^n h(L-1-n)$, with L representing the filter length which must be even and $0 \leq n \leq L-1$. The two functions work like a pair of filters (lowpass and highpass). The filters $h(n)$ and $g(n)$ is used as in the subband coding scheme described in chapter6. To reconstruct the input signal back from the filtered signals, the filtered signals are passed through inverse filters and added together. The inverse filters are identical to the filters themselves with time reversal, this again simplify the computation in $DTWT$. The $DTWT$ is used to create wavelet representation described in chapter 6.

Appendix B

Acknowledgements to Testing Images

Some testing image pairs used in this thesis are obtained from the VASC Image Database available on the Internet provided by Carnegie Mellon University's Vision and Autonomous Systems Center.

B.1 The Circuit Board Image

Descriptions
The images defined by and accompanying this header file may not be copied or redistributed without inclusion of this corresponding header file. This header file must include this message, and the acknowledgement of the source (see :SOURCE below) of the images.
(:TITLE "Apple Motherboard"
:FORMAT :RASTER
:FILENAME (:BEGIN "apple"
:END ".img"
:STEREO (:STEREO "l" "r")

:ORDER (:BEGIN - :STEREO :END))
:FILESIZE (:IMAGE-LINES 512
:PIXELS/LINE 512
:BITS/PIXEL 8)
:KEYS ("B/W"
"Circuit Board"
"Indoor"
"Stereo")

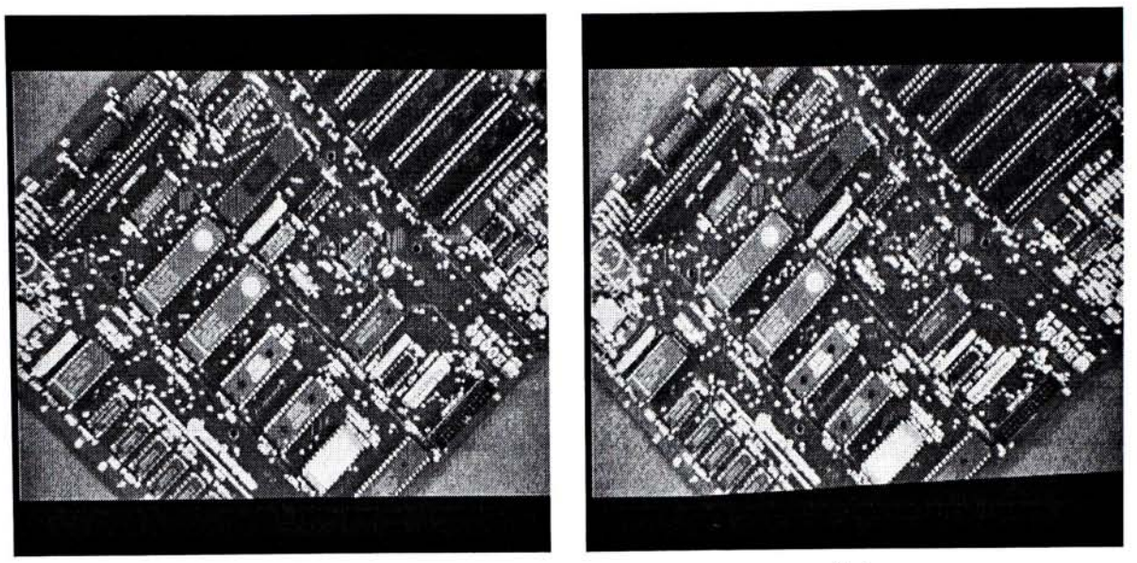
:SOURCE "University of Illinois, Bill Hoff"
:DESCRIPTION "Apple IIe motherboard, taken with TV camera."
:REFERENCES (:ARTICLE (
:AUTHOR "Hoff, W. and Ahuja, N."
:TITLE "Surfaces from Stereo: Integrating Feature Matching, Disparity Estimation, and Contour Detection"
:JOURNAL "IEEE Transactions on Pattern Analysis and Machine Intelligence"
:YEAR 1989
:MONTH "February"
:PAGES "121-136"
:VOLUME 11
:NUMBER 2)))

Images : (see fig. B.1)

B.2 The Stack of Books Image

Descriptions

The images defined by and accompanying this header file may not be copied or



(a) (b)

Figure B.1: The Circuit Board Images (512x512x256G); (a) left (b) right.

redistributed without inclusion of this corresponding header file. This header file must include this message, and the acknowledgement of the source (see :SOURCE below) of the images.

(:TITLE "Stack of Books"
:FORMAT :RASTER
:FILENAME (:BEGIN "books"
:END ".img"
:STEREO (:STEREO "l" "r")
:ORDER (:BEGIN - :STEREO :END))
:FILESIZE (:IMAGE-LINES 512
:PIXELS/LINE 512
:BITS/PIXEL 8)
:KEYS ("B/W"
"Book"
"Indoor"
"Stereo")

:SOURCE "University of Illinois, Bill Hoff"

:DESCRIPTION "Stack of books on table, taken with TV camera."

:REFERENCES (:ARTICLE (

:AUTHOR "Hoff, W. and Ahuja, N."

:TITLE "Surfaces from Stereo: Integrating Feature Matching, Disparity Estimation, and Contour Detection"

:JOURNAL "IEEE Transactions on Pattern Analysis and Machine Intelligence"

:YEAR 1989

:MONTH "February"

:PAGES "121-136"

:VOLUME 11

:NUMBER 2)))

Images : (see fig. B.2)

(a) (b)

Figure B.2: The Stack of Books Images (512x512x256G); (a) left (b) right.

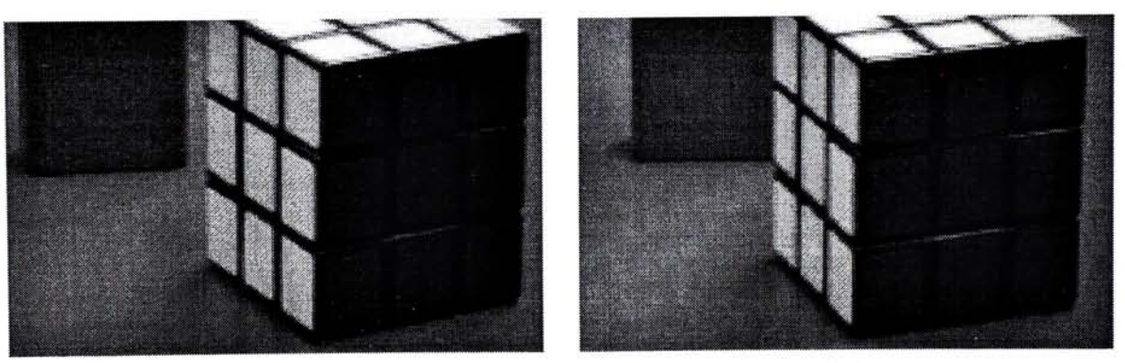
B.3 The Rubik Block Images

Descriptions

The images defined by and accompanying this header file may not be copied or redistributed without inclusion of this corresponding header file. This header file must include this message, and the acknowledgement of the source (see :SOURCE below) of the images.

(:TITLE "Toys Image, No. 3-2"
:FORMAT :RASTER
:FILENAME (:BEGIN "t3-2"
:END ".img"
:STEREO (:STEREO "l" "r")
:ORDER (:BEGIN - :STEREO :END))
:FILESIZE (:IMAGE-LINES 134
:PIXELS/LINE 212
:BITS/PIXEL 8)
:KEYS ("Blocks Scene"
"B/W"
"Stereo")
:SOURCE "USC Institute for Robotics and Intelligent Systems, Steven Cochran"
:DESCRIPTION "Partial view of a Rubik's cube occluding a wooden block.")

Images : (see fig. B.3)



(a) (b)

Figure B.3: The Original Rubik Block Images (212x134x256G); (a) left (b) right.

Bibliography

[AHUJ93] AHUJA, N. AND ABBOTT, A. L. Active stereo: Integrating disparity, vergence, focus, aperture, and calibration for surface estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1007–1029, 1993.

[AYAC87a] AYACHE, N. AND LUSTMAN, F. Fast and reliable passive trinocular stereo vision. In *Proc. 1st Int. Conf. Comput. Vision*, pp. 422–427, June 1987. London.

[AYAC87b] AYACHE, N. AND FAVERJON, B. Efficient registration of stereo images by matching graph descriptions of edge segments. *International Journal of Computer Vision*, 1(2):107–131, Spring 1987.

[BAKE81] BAKER, H. H. AND BINFORD, T. Deph from edge and intensity based stereo. In *Proc. Int. Conf. Artificial Intell.*, volume II, pp. 631–636, Aug. 1981.

[BALL82] BALLARD, D. H. AND BROWN, C. M. *Computer Vision.* Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[BURT83] BURT, P. J. AND ADELSON, E. H. the laplacian pyramid as a compact image code. *IEEE Transactions on Commun.*, COM-31:532–540, April 1983.

[CANN86] CANNY, J. F. A computational approach to edge detection. *IEEE Transactions on Pattern Anal. Machine Intell.*, PAMI-8(6):679–698, Nov. 1986.

[COCH89] COCHRAN, S. D. AND MEDIONI, G. Accurate surface description from binocular stereo. In *Proc. Workshop on Interpretation of 3D Scenes*, pp. 16–23, 1989.

[COHE89] COHEN, L., VINET, L., SANDER, P. T., AND GAGALOWICZ, A. Hierarchical region based stereo matching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 416–421, 1989.

[DAS95] DAS, S. AND AHUJA, N. Performance analysis of stereo, vergence, and focus as depth cues for active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1213–1219, 1995.

[DHON89] DHOND, U. R. AND AGGARWAL, J. K. Structure from stereo – A review. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1489–1510, Nov.-Dec. 1989.

[FAUG93] FAUGERAS, O. *Three-Dimensional Computer Vision - A Geometric Viewpoint.* The MIT Press, Cambridge, Massachusetts, 1993.

[GENN80] GENNERY, D. *Modelling the Environment of an Exploring Vehicle by Means of Stereo Vision.* PhD thesis, Computer Science Dept., Stanford University, Stanford, CA, 1980.

[GOLD89] GOLDSTEIN, E. B. *Sensation and Perception.* Wadsworth Publishing Company, Belmount, California, 1989.

[GRIM81] GRIMSON, W. E. L. A computer implementation of a theory of human stereo vision. *Phil. Trans. Royal Soc.*, 298:395–427, 1981.

[GRIM85] GRIMSON, W. E. L. Computational experiments with a feature based stereo algorithm. *IEEE Trans. Pattern Anal. Machine Intell.*, 7:17–34, 1985.

[GROS95] GROSSO, E. AND TISTARELLI, M. Active/dynamic stereo vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):868–879, Sept. 1995.

[HORA89] HORAUD, R. AND SKORDAS, T. Stereo correspondence through feature grouping and maximal cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(11):1168–1180, Nov. 1989.

[ITO86] ITO, M. AND ISHII, A. Three-view stereo analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 8:524–532, 1986.

[KAPP87] KAPPIE, F. AND LIEDTKE, C. E. Modeling of a natural 3-d scene consisting of moving objects from a sequence of monocular tv images. In *Proc. SPIE*, volume 860, pp. 126ff, 1987.

[KIM85] KIM, Y. C. AND AGGARWAL, J. K. Finding range from stereo images. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 289–294, 1985.

[KOCH93] KOCH, R. Dynamic 3-d scene analysis through synthesis feedback control. *IEEE Trans. Pattern Anal. Machine Intell.*, 15:556–568, 1993.

[LAW92] LAW, K. H. Progressive image transmission using wavelet multiresolution representation. Master's thesis, Dept. of Electronic Engineering, City Polytechnic of Hong Kong, 1992.

[LEUN94] LEUNG, C. W. AND WONG, K. H. Computer stereo vision using adaptive resonance theory. In *Proc. 3rd Int. Conf. on Automation, Robotics and Computer Vision*, pp. 192–196, Nov. 1994.

[LI94] LI, Z.-N. Stereo correspondence based on line matching in hough space using dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(1):144–152, Jan. 1994.

[MARR76] MARR, D. AND POGGIO, T. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

[MARR79] MARR, D. AND POGGIO, T. A computational theory of human stereo vision. *Proc. R. Soc. Lond.*, B204:301–328, 1979.

[MARR80] MARR, D. AND HILDRETH, E. Theory of edge detection. *Proc. R. Soc. Lond.*, B207:187–217, 1980.

[MARR82] MARR, D. *Vision - A Computational Investigation into the Human Representation and Processing of Visual Information.* W. H. Freeman and Company, San Francisco, 1982.

[MAYH81] MAYHEW, J. E. W. AND FRISBY, J. P. Psychophysical and computational studies towards a theory of human stereopsis. *Artificial Intell.*, 17:349–385, 1981.

[MORA80] MORAVEC, H. P. *Robot Rover Visual Navigation.* UMI Research Press, 1980.

[MOUS94] MOUSAVI, M. S. AND SCHALKOFF, R. J. Ann implementation of stereo vision using a multi-layer feedback architecture. *IEEE Transactions on System, Man, and Cybernetics*, 24:1220–1238, Aug. 1994.

[OHTA85] OHTA, Y. AND KANADE, T. Stereo by intra- and interscanline search using dynamic programing. *IEEE Transactions on Pattern Anal. Machine Intell.*, PAMI-7(2):139–154, Mar. 1985.

[OR91] OR, S. H. A cooperative algorithm for stereo disparity computation. Master's thesis, Dept. of Computer Science, The Chinese University of Hong Kong, Hong Kong, 1991.

[POLL85] POLLARD, S., MAYHEW, J., AND FRISBY, J. PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14, 1985.

[RIOU91] RIOUL, O. AND VETTERLI, M. Wavelets and signal processing. *IEEE Signal Processing Magazine*, 8(4):14–38, 1991.

[ROSE71] ROSENFELD, A. AND THURSTON, M. Edge and curve detection for visual scene analysis. *IEEE Transactions on Computer*, C-20(5):562–569, 1971.

[TOH90] TOH, P. S. AND FORREST, A. K. An edge and shading encoding and compression technique. In *Proc. International Conference on Automation, Robotics and Computer Vision*, pp. 983–987, 1990.