STUMP:

Stereo Correspondence in the Cyclopean Eye under Belief Propagation

by

George J. Distler

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
Department of Computer Science
Courant Institute of Mathematical Sciences
New York University
May 2008

Professor Davi Geiger

© George J. Distler

All Rights Reserved, 2008

To my parents and grandparents.

Acknowledgements

I would like to thank my thesis advisor, Professor Davi Geiger, for his encouragement, knowledge, and guidance. I am grateful for his endless patience and understanding taking the time to work through any problems or questions that I encountered throughout this project. I am also thankful for his sense of humor and philosophical conversations. I would also like to thank Professor Yann LeCun for being my second thesis reader and for his support in my efforts. I thank him for his patience and time. I would like to thank Professor Ken Perlin for providing useful ideas and sharing an interest, and also having a great sense of humor.

I am very grateful to the entire STUMP team. I would like to especially thank Kai Ju Liu for the contribution of his work on the BP-TwoGraphs algorithm, taking part in our discussions, and sharing his time with us. I thank him for his patience in explaining belief propagation and the mathematics behind it. I would like Arefin Huq both as a mentor and friend. Arefin provided the encouragement and support to embark on this thesis as well as gave advice ranging from programming to proofreading. I would also like to thank Fortunato Menezes for taking the time to be part of our STUMP project and for all the intriguing discussions we have shared. I would also like to thank all my friends who have supported me all along the way.

Finally, I would like to thank my family, especially my parents and grandparents who supported me though my education. Without them, this would not have been possible. I would like thank my father allowing me the opportunity to complete both a Bachelors and a Masters degree in Computer Science at New York University.

Abstract

The human visual system sees at any moment a static scene in three dimensions. This 3D view of the world is acquired by two images, one from the left eye and the other by the right eye. Fusing the left and right stereo pair of images yields a single cyclopean view portraying depth. Stereo vision can be applied to the field of computer vision via calibrated stereo cameras to capture the left and right images. Given a stereo pair of images, one can compute the field of depth via a stereo correspondence algorithm. We present a new approach to computing the disparity (depth) by means the STUMP algorithm.

The STUMP algorithm presents a solution to the stereo correspondence problem. We propose to solve the problem of discontinuities in disparity within epipolar lines by modeling geometric constraints of smooth, tilted, and occluded surfaces as well as unicity and opaqueness. Our algorithm runs upon a framework built upon the BP-TwoGraphs belief propagation estimation [17]. As a result, we provide a disparity map in the cyclopean coordinate system determined by a probability distribution computed in polynomial time.

Contents

Acknowledgements……...……iv

Abstract……vi

List of Figures…….....x

1 Introduction……..1

2 Previous Contributions……4

2.1 Projective Geometry & Epipolar Lines……..4

2.2 Bayesian Framework……14

2.3 Stereo Perception……16

2.4 Stereo Computation……..22

2.5 Bayesian Networks Computation (Belief Propagation)……...27

3 Stereo Correspondence……..38

3.1 The Matching Space……..38

3.2 Cyclopean Coordinate System…….41

3.3 Features……44

3.3.1 Multi-Scale Feature Integration and Entropy……49

3.4 Uniqueness – Opaque Constraint…….51

3.5 Smooth Surface Constraint……..53

3.5.1 Occluded, Tilted, and Flat Surfaces……..56

3.6 Modeling Surfaces……...61

3.6.1 Tilted Surfaces……..64

3.6.2 Occluded Surfaces…….65

3.6.3 Posterior Model…….68

3.7 Epipolar Line Interactions……69

3.8 Belief Propagation Implementation with BP-TwoGraphs……...73

4 Experiments and Results……...75

4.1 Settings……..76

4.2 Results……...81

5 Conclusion……115

Bibliography……..116

List of Figures

Figure 2.1.1: Projective Camera……6

Figure 2.1.2: Point Correspondence Geometry……....8

Figure 2.1.3: Epipolar Geometry……..10

Figure 2.1.4: Epipolar Lines with Two Projective Cameras……12

Figure 2.3.1: Random-Dot Stereogram……17

Figure 2.3.2: Illusory Square……18

Figure 2.3.3: Illusory Triangle……..19

Figure 2.3.4: Half-Occlusions……..20

Figure 2.5.1: A Pairwise Markov Random Field on a 5 by 4 Pixel Image……..29

Figure 2.5.2: BP-TwoGraphs Decomposed Graph Format……..30

Figure 2.5.3: Example of Computing a Belief……33

Figure 2.5.4: Horizontal Graph Message Propagation……..35

Figure 2.5.5: Vertical Graph Message Propagation……36

Figure 3.1.1: Matching Space……..…..39

Figure 3.1.2: Matching Space Graph……...40

Figure 3.2.1: Cyclopean Eye……43

Figure 3.3.1: Window Matching……..45

Figure 3.4.1: Uniqueness-Opaque Constraint……..52

Figure 3.5.1: Ordering Constraint……53

Figure 3.5.2: A Valid Surface and a Violation of the Ordering Constraint…….55

Figure 3.5.3: Flat Plane versus Double Tilted Plane…….57

Figure 3.5.4: Flat Plane versus Occluded Plane……...59

Figure 3.5.5: Flat Plane versus Occluded Plane (Color)……..60

Figure 3.5.6: Flat and Occluded Planes versus Tilted Planes……..61

Figure 3.6.1: Tilted Surface Model……..64

Figure 3.6.2: Occluded Surface Model……67

Figure 3.7.1: Stereo Pair of an Illusory Rectangle……...69

Figure 3.7.2: Vertical Epipolar Line Interaction……..70

Figure 3.7.3: Modeling Disparity across Epipolar Lines……71

Figure 4.1.1: Cyclopean Disparity Map as Seen From the Left and Right Eyes…….80

Experiment 1.1: Random-Dot Stereogram……...83

Experiment 1.2: Illusory Square……..84,85

Experiment 1.3: Pentagon……...86,87

Experiment 1.4: Cube……...88

Experiment 1.5: Fruit……..89,90

Experiment 2.1.1: Random-Dot Stereogram scale……...91

Experiment 2.1.2: Illusory Square scale……..92

Experiment 2.1.3: Pentagon scale……...93,94

Experiment 2.1.4: Cube scale……..95

Experiment 2.2.1: Random-Dot Stereogram η……...96

Experiment 2.2.2: Illusory Square η……...97

Experiment 2.2.3: Cube η……..98,99

Experiment 2.3.1: Random-Dot Stereogram μ ……..99

Experiment 2.3.2: Illusory Square μ ……100

Experiment 2.3.3: Pentagon μ ……..101

Experiment 2.3.4: Cube μ ……..102,103

Experiment 2.4.1: Random-Dot Stereogram *tilt* ……...103

Experiment 2.4.2: Illusory Square *tilt* ……...104

Experiment 2.4.3: Cube *tilt* ……...105

Experiment 2.5.1: Random-Dot Stereogram *occlusion* ……106

Experiment 2.5.2: Illusory Square *occlusion* ……107

Experiment 2.5.3: Cube *occlusion* ……..107,108

Experiment 3.1: Random-Dot Stereogram.……..109

Experiment 3.2: Illusory Square.……110

Experiment 3.3: Pentagon……..111

Experiment 3.4: Cube.……112

Experiment 3.5: Fruit……..113,114

1 Introduction

Binocular vision, known as stereopsis, is the process in which the visual sense obtains a perception of depth from a pair of left and right images. Stereoscopic depth is the sensation that emerges from the fusion of two slightly different views of the world on the two retinas. A sense of depth can be obtained by a pair of images either biologically or via a pair of cameras. The difference between the two views as seen by both the left and right eyes is referred to as binocular disparity. Binocular disparity occurs from the result of the horizontal separation of the left and right eyes. The discovery that binocular disparity is interpreted as depth by the brain was published by Charles Wheatstone in his significant paper to the Royal Society in 1838 [32].

Leonardo da Vinci also realized that objects in the world, at different distances from the eyes, project an image in the left eye and an image in the right eye which differ in their horizontal positions. However, he concluded that it was impossible for a painter to portray a realistic depiction of depth in a scene with a single canvas.

The question of obtaining a single notion of depth from a pair of views seen by the eyes can be translated into the realm of computer vision with the use of a pair of specially calibrated cameras. These cameras will produce a left and right stereo pair

of images similar to the views produced by each eye. Given a stereo pair of images, the first task is to detect features which occur in both images. Next, one must develop a correspondence between the features of the stereo pair of images. Finally, depth can be computed by measuring the difference between corresponding feature matches with the stereo pair.

Our approach, takes into account the problem of stereo correspondence by utilizing the cyclopean coordinate system, which is a system that represents a single fused view of the stereo pair of images. The cyclopean view was chosen since it is a natural way to view depth. Depth is seen only through a single view fused in the brain by images from both eyes. The STUMP algorithm will contribute to a procedure for conducting stereopsis in computer vision.

The foundation of stereopsis became popular with the modeling of image projections in cameras. Chapter 2 will provide a mathematical model necessary for stereo vision consisting of both projective and epipolar geometry. We will also discuss the Bayesian framework, which yields a probability distribution based on parameters using a prior model of images. Chapter 2 will also introduce stereo perception, which explains how the world is perceived in three dimensions independent of object recognition as well as mentioning the occluded regions within a stereo pair of images, which effects the stereo matching. Next we will introduce the current state of stereo computation in the computer vision field and the current

algorithms used for stereo correspondence. Concluding Chapter 2 will be a discussion on belief propagation as well as the BP-TwoGraphs estimation contributed by Kai Ju Lui [17].

Our goal is to provide an estimation of depth in the cyclopean view from the images seen by the left and right eyes. In Chapter 3, we will introduce our solution to stereo correspondence, the STUMP algorithm, by first defining the problem in terms of the stereo matching space and cyclopean coordinate system. We will provide a detailed explanation of the extraction of features from a stereo pair of images, as well as how we model various types of surfaces and how we determine the disparity of pixels within the pair of images.

After our theory of stereo correspondence is discussed in Chapter 3, we will illustrate and analyze the STUMP algorithm's performance on various stereo images in Chapter 4. This will include a comparison to an optimal solution of solving disparity for independent horizontal (epipolar) lines via a dynamic programming algorithm. We will show that the STUMP algorithm provides a solution to stereo correspondence utilizing marginal probabilities of local disparities.

2 Previous Contributions

The STUMP algorithm contributes to the computer vision field with a method to solve the stereo vision correspondence problem. Prior to the work of the STUMP algorithm, many well-known ideas have been studied and proved quite useful in the development and experimentation of the algorithm. The mathematical geometry used to model stereo images dates back to the pinhole camera. Built upon the projective camera model, stereo vision takes into account a pair of cameras and the epipolar geometry used in matching the corresponding points between the stereo pair of images.

2.1 Projective Geometry & Epipolar Lines

The intrinsic geometry for stereo vision is epipolar geometry, which is the geometric foundation for the projection of an image in 3D space onto two viewing planes. Under the assumption that two cameras (image planes) can be well approximated by the pinhole camera model, each camera captures a projected image of the 3D world. In stereo vision, the correspondence problem is the reconstruction of

the 3-D coordinates of a number of points in a scene given several images (two in our case) obtained by cameras of known relative positions and orientations.

To understand how points in the 3D world are projected onto 2D images, one must understand the projective camera model. In Figure 2.1.1 below, let $P_O = (X, Y, Z)$ where a point in the 3D world is represented by a "world" coordinate system. Let O be the center of projection of a camera where a camera reference frame is placed. The camera coordinate system has the Z component perpendicular to the camera frame (where the image is produced) and the distance between center O and the camera frame is the focal length, f. In this camera coordinate system, the point $P_O = (X, Y, Z)$ is described by the vector $\vec{P}_O = (X_O, Y_O, Z_O)^T$ and the projection of this point onto the image is given by the point $\vec{p}_O = (x_O, y_O, f)^T$ where $\vec{p}_O = \frac{f}{Z}\vec{P}_O$.

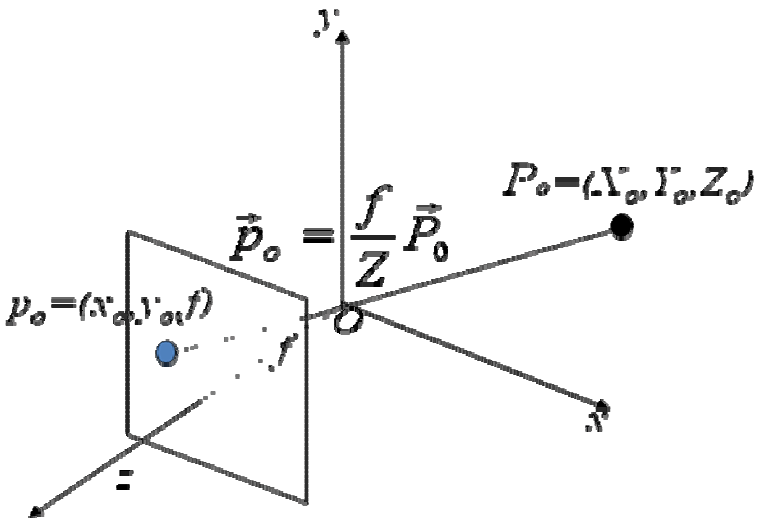


Figure 2.1.1: Projective Camera. The figure above depicts the projection of the point $P_O = (X_O, Y_O, Z_O)$ in the 3D world coordinate system (black) onto the image 2D coordinate system $p_O = (x_O, y_O, f)$ (blue).

Within the projective camera coordinate system, let $x_0 = (q_{x0} - o_x)s_x$ and $y_0 = -(q_{y0} - o_y)s_y$ where the intrinsic parameters of the camera are $\{(s_x, s_y), (o_x, o_y), f\}$. Let (s_x, s_y) represent the size of the pixels along the x and y directions. Let (o_x, o_y) represent the coordinate in pixels of the image and f represents the focal length of the camera. This conversion between the camera coordinate system to the image coordinate system can be described by the following linear transformation. This linear transformation is necessary due to the fact that the points in the camera coordinate system are continuous. However, in the image

coordinate system, the points are discrete pixel values. The following equations display the conversion from a vector $\vec{p}_o$ in the camera coordinate system, to a vector $\vec{q}_o$ in the image coordinate system and vice versa.

$$\vec{p}_o = Q^{-1}\vec{q}_0 \qquad (2.1.1)$$

$$\vec{q}_o = Q\vec{p}_0 \qquad (2.1.2)$$

$$Q^{-1} = \begin{pmatrix} s_x & 0 & -s_x o_x \\ 0 & -s_y & s_y o_y \\ 0 & 0 & f \end{pmatrix} \qquad (2.1.3)$$

$$Q = \begin{pmatrix} \frac{1}{s_x} & 0 & \frac{o_x}{f} \\ 0 & -\frac{1}{s_y} & \frac{o_y}{f} \\ 0 & 0 & \frac{1}{f} \end{pmatrix} \qquad (2.1.4)$$

In stereo vision, two projective cameras are used simultaneously to take two 2D images of the same 3D scene. Figure 2.1.2 as shown below, adapted from Zisserman [11], depicts a pair of stereo images and a point X in 3D space. Points X_{LEFT} and X_{RIGHT} are the projections of the point X onto the left and right image

planes. The image points X_{LEFT}, X_{RIGHT}, X, and the centers of cameras C_{LEFT} and C_{RIGHT} (image planes) are all coplanar. This plane is known as the epipolar plane, denoted by π.

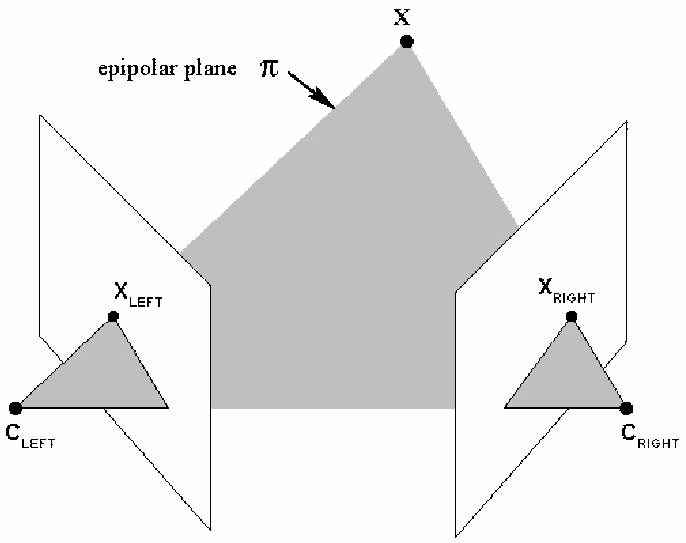


Figure 2.1.2: Point Correspondence Geometry. The two cameras are indicated by their centers C_{LEFT} and C_{RIGHT} and image planes. The camera centers, the 3D point X, and its images X_{LEFT} and X_{RIGHT} lie in a common plane π. Note that the centers of the cameras are behind the image planes. This image has been adapted from Zisserman [11].

Under the assumption that we only know X_{LEFT}, we wish to know how the corresponding point X_{RIGHT} is constrained. The following figure below depicts the epipolar geometry which yields the corresponding epipolar lines between two images. The plane π is determined by the baseline and the ray defined by X_{LEFT}. The baseline

is the line along the epipolar plane π, which connects points C_{LEFT} with C_{RIGHT}. From Figure 2.1.2 above, we know that the ray corresponding to the unknown point X_{RIGHT} lies in π; therefore the point X_{RIGHT} lies on the line of the intersection l' of π with the second image plane. This line l' is the image in the second view of the ray back-projected from X_{LEFT}. Thus, the line l' is the epipolar line corresponding to X_{LEFT}. An epipolar line is the intersection of an epipolar plane with the image plane. All epipolar lines intersect at the epipole. An epipolar plane intersects the left and right image planes at epipolar lines, and defines the correspondence between the lines. In relation to a stereo correspondence algorithm, the benefit is that the search for the point corresponding to X_{LEFT} does not need to cover the entire image plane but can be restricted to the line l'.

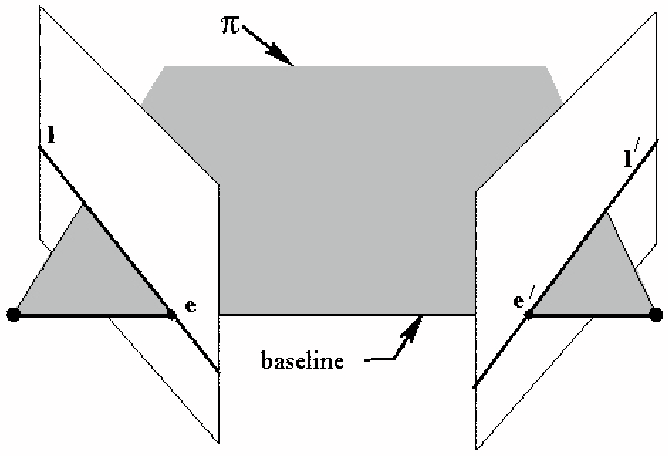


Figure 2.1.3: Epipolar Geometry. The camera baseline intersects each image plane at the epipoles e and e'. Any plane π containing the baseline is an epipolar plane and intersects the image planes in corresponding epipolar lines l and l'. This figure has been adapted from Zisserman [11].

Since the epipolar line l' is the projection in the second image of the ray from the point X_{LEFT} through the camera center C_{LEFT} of the first camera, there is a mapping $x \mapsto l$ from a point in one image to its corresponding epipolar line in the second image. This mapping from points to lines is represented by the fundamental matrix F. The fundamental matrix is responsible for estimating the epipolar lines. In a stereo system consisting of two projective cameras, each image plane can be thought of as a rotation and translation of the other which is computed by the fundamental matrix.

Consider the following diagram in which a single point $P = (X, Y, Z)$ in the 3D world is projected into two projective cameras (stereo pair of image planes). The

transformation of the coordinate system from the left camera into the right camera is described by a rotation matrix R and a translation vector T. This means that a point $P = (X, Y, Z)$ described as $\vec{P}_l$ in the left frame will be described in the right frame as $\vec{P}_r = R^{-1}(\vec{P}_l - \vec{T})$. Each 3D point $P = (X, Y, Z)$ defines a plane $\overline{PO_lO_r}$, which is shown as the blue triangle in the figure below. This plane intersects the left and right camera frames creating two epipolar lines denoted by e_l and e_r respectively. The line $\overline{O_lO_r}$ will intersect the camera planes at e_l and e_r. These two intersection points are known as the epipoles and are usually outside of the image window of the camera plane. They are not depicted in the diagram below. The line $\overline{O_lO_r}$ is common to every plane $\overline{PO_lO_r}$ and therefore the two epipoles belong to all pairs of epipolar lines. The epipoles can be thought of as the intersection of all epipolar lines.

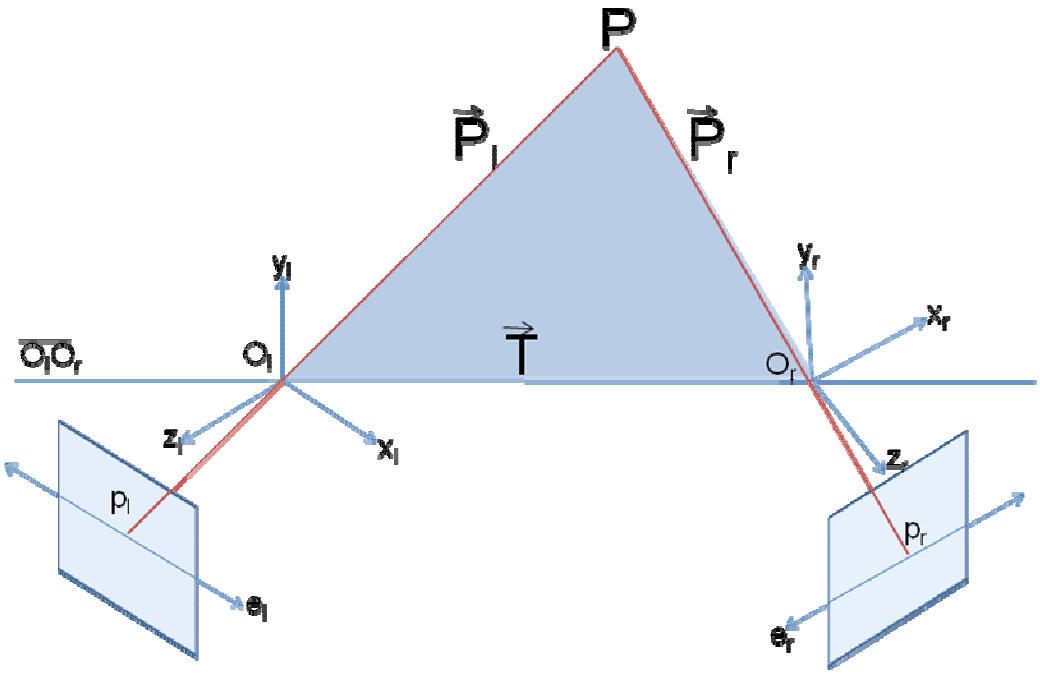


Figure 2.1.4: Epipolar Lines with Two Projective Cameras. The diagram above depicts a 3D point $P=(X,Y,Z)$ onto both image (camera) planes denoted by the light blue planes. The point P is projected onto the point $p_l=(x_o,y_o,f)$ in the left image plane by vector $\vec{P}_l$ (red) and is projected onto the point $p_r=(x_o,y_o,f)$ in the right image plane by vector $\vec{P}_r$ (red). Each image plane contains its relative coordinate system. The left image is denoted as $\{\vec{x}_l,\vec{y}_l,\vec{z}_l\}$ and the right image as $\{\vec{x}_r,\vec{y}_r,\vec{z}_r\}$ with origins denoted as O_l and O_r respectively. The blue triangle defines the plane of the line $\overline{O_lO_r}$ and point P. This plane intersects the two image planes creating the two corresponding epipolar lines e_l and e_r.

Since the two vectors, $(\vec{T},\vec{P}_l)$, span a 2D plane, their cross product $(\vec{T}\times\vec{P}_l)$ is perpendicular to this plane. As a result we have the following equation (2.1.5). This equation states that a point in the left coordinate system can be written as a point in the

right coordinate system and vice versa by performing a rotation and translation, equation (2.1.6).

$$(\alpha \vec{P}_l - \beta \vec{T})^T \bullet (\vec{T} \times \vec{P}_l) = 0 \qquad (2.1.5)$$

We can substitute $\gamma = \frac{\alpha}{\beta}$, yielding $\beta(\gamma \vec{P}_l - \vec{T})^T \bullet (\vec{T} \times \vec{P}_l) = 0$

By substituting $\vec{P}_l' = \gamma P_l'$ we get $\beta(\gamma \vec{P}_l' - \vec{T})^T \bullet (\vec{T} \times \vec{P}_l) = 0$

We can then define $\vec{P}_r = R^{-1}(\vec{P}_l - \vec{T})$ and get $\beta(R\vec{P}_r')^T \bullet (\vec{T} \times \vec{P}_l) = 0$

We can then reduce to $\beta \vec{P}_r'^T R^T S(T) \vec{P}_l = 0$ where $S(T) = \begin{pmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{pmatrix}$

By letting $E(R,T) = R^T S(T)$, known as the essential matrix, we can further reduce to $\vec{p}_r'^T E(R,T) \vec{p}_l = 0$ (2.1.6)

At this moment, points $\vec{p}_r'^T$ and $\vec{p}_l$ are coordinates in the camera coordinate system will then need to be transformed into coordinates within the image coordinate system via the matrices Q and Q^{-1} as previously defined in equations (2.1.3, 2.1.4). We can now write equation (2.1.6) using the camera to image transformations

as $\vec{q}_r'^{\,T} Q_r^{-T} E(R,T) Q_l^{-1} \vec{q}_l = 0$. As a result we can derive the fundamental matrix F as:

$$\vec{q}_r'^{\,T} F(R,T,i_l,i_r)\vec{q}_l = 0 \qquad (2.1.7)$$

The fundamental matrix F can be estimated using the "Eight Point Algorithm." This algorithm states that given two images one must identify eight points or more on both images that are non-degenerate meaning that $n \geq 8$ points with their correspondence are provided. Then there are n linear and homogeneous equations $\vec{q}_r'^{\,T} F(R,T,i_l,i_r)\vec{q}_l = 0$ with 9 unknowns which are the components of F. Since we can only estimate F up to some scale factors there are only 8 unknowns to be computed from the $n \geq 8$ linear and homogenous equations. If $n = 8$ then there is a unique solution with non-degenerate points. If $n > 8$ then the solution is over-determined and Singular-Value-Decomposition can be used to find the best fit solution.

2.2 Bayesian Framework

The Bayesian Framework is an approach to perform analysis by synthesis. The goal is to generate a probability model for the disparity values for a given stereo pair of cameras and images. This can be thought of as a disparity field $d(e,x)$, where (e,x,z) belongs to a chosen coordinate system, where a reference camera has the

coordinates (e,x) with z representing disparity values so that the solution can be described as $(e,x,d(e,x))$. If we have such a probability model defined as $P(d \mid I^L, I^R)$, we can then ask: what is the most likely disparity field or what is the expected value for the disparity field? More generally, we can make any estimation of $d(e,x)$ given the probability $P(d \mid I^L, I^R)$. Usually, the maximum a posteriori probability estimator (MAP estimator) is considered, which provides the solution that maximizes the probability model. There are other estimators that are also of interest, such as the mean posterior probability estimator (MPM). Let us now consider the model in more detail. The probability to be derived is:

$$P(d \mid I^L, I^R)$$

The Bayes rule states: $$P(d \mid I^L, I^R) = \frac{P(I^L, I^R \mid d)\, P(d)}{P(I^L, I^R)}$$

Thus, the synthesis generates a model of the images from the disparity field

$$P(I^L, I^R \mid d)$$

and a model of the preference (bias) of the disparity field

$$P(d)$$

The model of the bias for the images, which is the denominator of the right hand side of the Bayes formula, does not need to be derived since it can be obtained as the normalization value. This formulation allows us to focus on developing both the synthesis model and the bias model in order to produce a model for stereo vision.

2.3 Stereo Perception

The perception of depth is the visual ability to perceive the world in three dimensions. Stereo perception is allowing one to gauge how distant an object is with high accuracy through triangulation with a pair of stereo images. In 1838, Charles Wheatstone's discovery of the stereoscope first demonstrated that disparity gives rise to stereopsis which is the process in visual perception leading to stereoscopic depth. Stereopsis itself gives us the experience on relative depth. It enables us to rank-order the nearness-farness of object within a region of space around a fixation point. Recent innovations in stereo perception include work done on random-dot stereograms, the cyclopean view, as well as considerations of occluded objects.

In 1959, Bela Julesz was responsible for the introduction of random-dot stereograms which contributed to new insights into binocular depth perception. Random-dot stereograms made it possible to portray images or events binocularly or binaurally that do not exist even physically on the left and right retina or cochleae

[15]. The random textures used deprive the subject of any familiarity cues and these stimuli individually do not contain any global information on the retinas [15]. Only at some central level on the cyclopean retina can the two retinal images be combined to portray a cyclopean image [15]. Stereopsis is based on the geometrical fact that two-dimensional projects of a three-dimensional object on the left and right retinas differ in their horizontal positions. This horizontal shift between corresponding points in the two retinal images is called retinal disparity (or just disparity). The following Figure 2.3.1 depicts a random-dot stereogram. The images both have the same random distribution of pixels however in one image a centered square is displaced by 4 pixels. Fusing the two images yields the appearance of a square emerging into 3D space.

Figure 2.3.1: Random-Dot Stereogram. The left image is generated by randomly assigning black and white values at each pixel. The right image is generated by copying the left image, however an imaginary square inside the left image is displaced 4 pixels to the left and the empty space is filled with random black and white values. Fusing the two images shows a square in front of the background.

With random dot stereograms, each image individually shows only a random distribution of pixels. Only by fusing a pair can one recognize an object in the images. This shows that obtaining depth from vision is a separate independent process from object recognition. This statement can be supported further by considering illusory contours. Stereo matching occurs even in the presence of an illusion. Even if the illusory contour is unknown, stereo matching can still occur. As shown in Figure 2.3.2 below, each stereo pair of images yields the presence of an illusory from stereo matching whether or not illusory figures of the stereo pair even match.

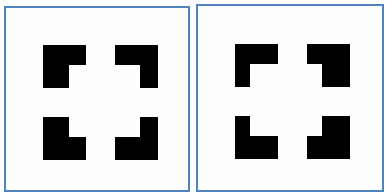


Figure 2.3.2: Illusory Square. The above pair of images contains 4 black squares in the background with a white square in the foreground displaced 4 pixels. Fusing the two images yields a single white square in the foreground with 4 black squares in the background.

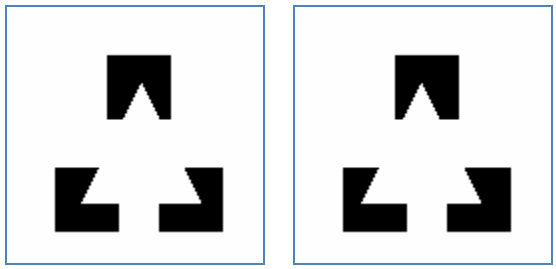


Figure 2.3.3: Illusory Triangle. The above pair of images contains 3 black squares in the background and a white triangle in the foreground. Fusing the two images yields a white triangle in the foreground and 3 black squares in the background.

These pairs of images provide evidence that the human visual system does not process illusory contours and surfaces before processing binocular vision. One can also conclude that binocular vision is a process that does not require any recognition or contour detection a priori.

The concept of occlusions dates back to half-occlusions noted by Leonardo da Vinci. Half-occlusions are regions of a left image that will have no match in the right image and vice-versa. Unmatched regions contain important information about the reconstruction of the scene. Although these regions can be small, they affect the overall matching scheme since the remaining matching must reconstruct a scene that accounts for the half-occlusion.

Leonardo da Vinci noted that the larger the discontinuity between two surfaces is, the larger the half-occlusion. Recent work, done in 1991 by Nakayama and

Shimojo [21], has shown in the following figure that inducing occlusions in a stereo pair of images affects the overall matching of the stereo pair. This is done by adding an extra dot to one of the images of the stereo pair.

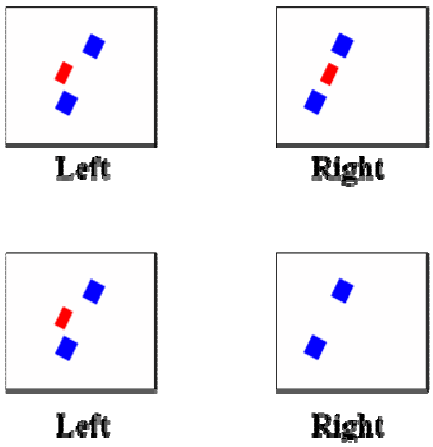


Figure 2.3.4: Half-Occlusions. The top stereo pair of images depicts two shifted blue dots yielding that the red dot is in the background and that the blue dots are in the foreground. The bottom stereo pair of images is identical to the top pair with the exception that the red dot of the right image is missing. The result of fusing the bottom pair also yields the red dot in the background and the blue dots in the foreground even though the data for the red dot is missing in the right image.

The work on occlusions, done by Nakayama and Shimojo [21], shows that unpaired points can be perceived at the appropriate depth if they are near fused targets and the eye-of-origin is appropriate. Unpaired points can also lead to the formation of subjective occluding contours seen in front. Eye-of-origin information is used to distinguish which side of the occluded object should be in the back or the front. There

are two ways of using eye-of-origin information for depth, edge and surface perception. Wheatstone's stereopsis is based on the signed difference in retinal disparity while da Vinci's stereopsis is based on the differential occlusion of image points.

While occlusions and illusory figures stir up quite an interest in stereo perception, one cannot neglect the impact of the cyclopean view in the area of stereo perception. Aside from his contribution of the random-dot stereograms, Bela Julesz also pioneered the concept of the cyclopean eye. Inspired by the mythical one-eyed Cyclops, Hering and Helmholtz used the term "cyclopean" to refer to a hypothetical single "eye", the "mind's eye", that sees a single stereoscopic image given two appropriate stimuli in the two eyes [15]. With random-dot stereograms it is possible to portray information on the "mind's retina" – that is, at a place where the left and right visual pathways combine in the visual cortex [15]. The cyclopean view is considering each point of the stereo images as a single matching point in 3D space. It is the view of fusing the two images together to see one image. Our algorithm addressed in the presented paper is based upon the cyclopean coordinate system (addressed in Chapter 3) which is constructed from the concept of mapping a both the left eye and the right eye to a single cyclopean viewpoint. This single-eye viewpoint seems to be perhaps the natural way of visioning the world in 3D.

2.4 Stereo Computation

Previous efforts on stereo computation include the concepts of cooperative stereo, disparity gradient, and such stereo algorithms as max flow and dynamic programming. Marr and Poggio [20] paved the way for the majority of the work to be done on stereo computation. They presented and defined the problem of extracting disparity information from a stereo pair of images. From this, they also derived a cooperative algorithm to solve for disparity.

Cooperative algorithms operate on many input elements and reach a global organization by means of local and interactive constraints. The term "cooperative" refers to the way in which local operations appear to cooperate in forming global order in a well-regulated manner [20]. While the cooperative concept was already well-known in other disciplines, its importance began to rise in the field of stereo vision. Perhaps one of the earliest suggestions similar to cooperative stereo was made by Julesz, who claimed that stereoscopic fusion is a cooperative process [15]. Marr and Poggio [20] defined the problem of measuring disparity as follows. First, a particular location on a surface in the scene must be selected from one image. Second, that same location must be identified in the other image. Finally, the disparity in the two corresponding image points must be measured. Defining the problem as stated above seems at first to be trivial, however the problem lies in identifying a locations beyond doubt in the two images and then the problem would be reduced to a problem of

measurement. Therefore the question of correspondence is the issue. In order to formulate the correspondence computation correctly, one must examine the physical world. Marr and Poggio [20] identified two constraints of importance. First, a given point on a physical surface has a unique position in space at any one time. Second, matter is cohesive and is separated into objects, and the surfaces are generally smooth compared with their distance from the viewer. These constraints apply to locations only on the physical surface. Therefore, in the computation, we must ensure that the items to which they belong to in the image are in one-to-one correspondence with well-defined locations on a physical surface.

After defining the necessary constraints for correspondence, the left and right descriptions for each eye can be combined using the concepts of uniqueness and continuity [20]. The concept of uniqueness declares that each item from each image may be assigned at most one disparity value. This condition relies on the assumption that an item corresponds to something that has unique physical position. The concept of continuity states that disparity varies smoothly almost everywhere. This condition is a consequence of the cohesiveness of matter, and it states that only a small fraction of the area of an image is composed of boundaries that are discontinuous in depth. Early on, Marr and Poggio [20] presented a cooperative algorithm to solve the correspondence problem. Over time the algorithms have evolved to state-of-the-art algorithms such as dynamic programming and later max flow.

A dynamic programming approach to the stereo correspondence problem was presented by Ohta and Kanade [22] in 1985. Their dynamic programming algorithm consisted of intra-scanline (finding a matching path on a 2D search plane) and inter-scanline (search in a 3D space in which is a stack of the previously solved 2D search planes) searches. Our STUMP algorithm presented in this paper will make comparisons to results of a dynamic programming solution. We also implemented a dynamic programming algorithm similar to the one presented by Ohta and Kanade to test our STUMP algorithm against. These results will be presented and discussed in the following chapters.

Currently the state-of-the-art stereo correspondence algorithm is the max flow / min-cut algorithm developed first by Cox [26], and improved in Boykov [4] and Ishikawa & Geiger [12]. The max flow algorithm requires that the prior model be described by a convex function on first order derivatives of the disparity [12]. The max flow algorithm has the interesting property that it provides the global optimal disparity value, taking into account the interactions among epipolar lines. We analyze the differences between the STUMP method and max flow, although we did not implement the max flow algorithm to compare our result to. Instead we will discuss the advantages of STUMP over max flow. It is worth noting that we do investigate the differences of the STUMP and max flow for the test images in common, based on the results of the max flow algorithm as presented by Ishikawa and Geiger [12].

Perhaps the quintessential difference between STUMP and max flow is that the STUMP algorithm utilizes belief propagation to assign disparity values based on the probabilities and retains the probability distribution over the disparity values, while the max flow only assigns a single (optimal) disparity value. The advantages of having the probability distribution of disparities, is that it allows for a better integration with other computer vision models that may have extra information regarding depth. For example, motion is an excellent well-known module used to detect occlusions and the relative depth of objects. Object recognition is another model that may contribute to the final depth solution. The combination of probability information is more natural since it contains more information. Specifically, it contains the score information (probability) for any disparity solution. Therefore, different solutions can be ranked. With the max flow solution, we only have the optimal solution and cannot have information about any other solution other than the fact that any other solution is not optimal. These differences in approach cannot be test here, since we are not focusing on the integration aspects of stereo. However, it is important to take into account the design of stereo modules. In the past, when researches devised algorithms for edge detection, there was a great emphasis placed on obtaining the "best" edge detector, in which decisions were placed at each pixel to detect if an edge was present. Canny's edge detector [6] was optimal according to a signal to noise ratio criteria. Currently, it is more common to apply "edge detection" algorithms that simply return a pixel's response to contrast rather than obtain a "best"

decision on whether a pixel is an edge or not. Accordingly, we believe that returning probability as the output of a stereo correspondence algorithm is more of a complete answer to the stereo matching problem.

Another advantage of choosing the belief propagation technique is that it allows us to model discontinuities, both horizontal and vertical, in more flexible ways. In the max flow approach, the cost function must be convex [12] and thus, the penalty for disparity discontinuities must be a convex function on the disparity difference. However, using the belief propagation method, discontinuities can be modeled arbitrarily. There is no reason to greater penalize vertical changes in disparity if the changes are larger. Depth discontinuities occur at object boundaries and different penalties should not be placed if objects are closer or further away from the background. Along the epipolar axis more considerations should be placed. However, we prefer a framework where the freedom to choose penalty functions is offered. This is the Bayesian belief propagation framework.

These two reasons, outputting a probability solution for all disparity values and being flexible on the choice of penalties, are the main reasons advocating the belief propagation method. However, one drawback of the Bayesian belief approach is that we cannot guarantee an optimal solution. Nevertheless, studies by researches in Bayesian belief propagation methods offer reasonable assurance that the method works quite well, and therefore we chose that method.

In particular, our algorithm uses the trainable graph combination scheme for belief propagation as developed by Kai Ju Liu [17]. This belief propagation scheme, called BP-TwoGraphs, will be presented as follows.

2.5 Bayesian Networks Computation (Belief Propagation)

Belief propagation in general, has strong limitations such as NP hard solutions, expensive computations, and no guarantee of convergence as well as being difficult to implement [17]. Instead, the graph combination scheme is used in providing a highly accurate and efficient means for belief propagation.

The graph combination scheme is comprised of a given Markov random field approximated by multiple sets of singly-connected graphs. The exact belief propagation on each set is then combined to yield a final set of marginal values. The non-linear combination of information from each set of graphs improves on the performance of each set alone while yielding accuracy approaching the optimal produced by the max flow algorithm.

In computer vision, many problems including stereo correspondence can be expressed in terms of probability theory under a Bayesian framework. A space of the parameters to be estimated is defined, and the model of the relationship between the parameters and the resulting image is given as a conditional probability distribution.

By means of Bayes' theorem the reverse relationship can be deduced from the values of the parameters that occur a priori. For a given image, the parameters are estimated by maximizing this probability. Since we are often interested in a single parameter and consider the others to be unknown, it becomes natural to marginalize the remaining parameters by summing over all of their possible values, resulting in the marginal probability of the parameter that we wish to estimate. These marginal probability calculations are the foundation of belief propagation since the computed beliefs are the marginal probabilities.

Traditional belief propagation consists of a pair-wise Markov random field which is a graph of vertices connected by edges. As can be seen in the Figure 2.5.1 below, each vertex has an observed value y and a set of possible states X, which are denoted for vertex i by y_i and X_i. The objective is to infer information about the states $x_i \in X_i$ given the observed y_i. In order to take advantage of a Markov random field, there must be a relation between a vertex's states and its observed value as well as a relation between neighboring vertices. This being said, there is an assumption that there is a statistical dependency between y_i and $x_i \in X_i$ given a non-negative compatibility function $\phi_i(x_i, y_i)$. The relationship between neighboring vertices i and j is given by another non-negative compatibility function $\psi_{ij}(x_i, x_j)$, where $x_i \in X_i$ and $x_j \in X_j$. The function ψ is also symmetric, meaning that $\psi_{ij}(x_i, x_j) = \psi_{ji}(x_j, x_i)$. Computing the beliefs at each vertex for each given state

by propagating messages from neighboring vertices and states can be NP hard in a loopy graph. However, the message computations needed for singly-connected-graphs terminate at the leaf vertices allowing beliefs to be computed exactly and efficiently.

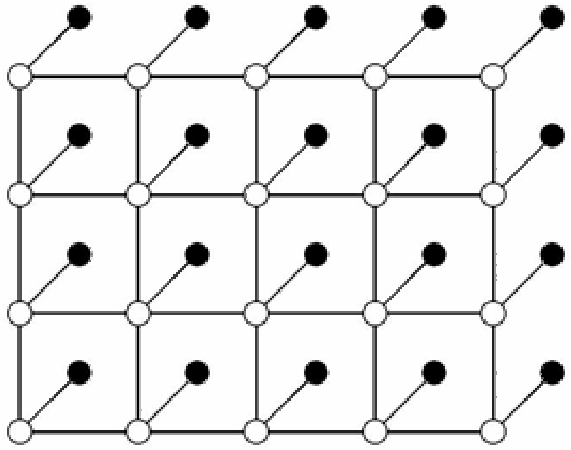


Figure 2.5.1: A Pairwise Markov Random Field on a 5 by 4 Pixel Image. Each vertex corresponds to an image pixel. Each vertex is connected to its horizontal and vertical surrounding neighbors. The filled circles represent the observed values y_i and the empty circles represent the states X_i. This figure is borrowed from Kai Ju Liu [17].

Due to the difficulties in solving belief propagation for loopy graphs, the BP-TwoGraph algorithm was presented which combines belief propagation on a collection of singly-connected graphs. The BP-TwoGraphs algorithm combines the information from multiple singly-connected graphs to produce an estimation that is comparable to belief propagation on the loopy graphs in an efficient manner with a

fraction of the complexity. The BP-TwoGraphs algorithm captures the connections and loops of the original graph of the Markov random field by decomposing the graph into multiple singly-connected graphs in the form of rectangular grids. The selection of the dual sets of complementary horizontal and vertical graphs, as shown in Figure 2.5.2 below, have been demonstrated to be highly effective [17].

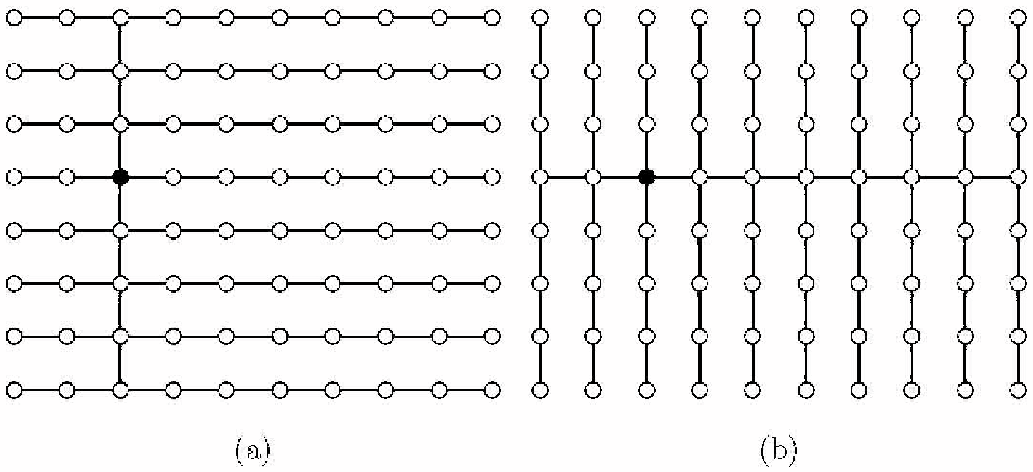


Figure 2.5.2: BP-TwoGraphs Decomposed Graph Format. The (a) horizontal graph and (b) vertical graph for the filled pixel vertex of interest is shown above. These figures are borrowed from Kai Ju Liu [17].

The above figure shows the complimentary singly-connected-graphs for a given pixel vertex. By combing both the horizontal and vertical graphs, the pixel vertex of interest is connected to all other pixel vertices in the horizontal graph along all rows horizontally and vertically along its column (as shown in figure 2.5.2 (a)).

Similarly, in the vertical graph the pixel of interest is connected to all other pixel along all columns vertically and horizontally along its row (as shown in figure 2.5.2 (b)). In order to combine the two graphs, belief propagation is applied to both graphs to generate a set of beliefs for each node and state for each graph. The final set of beliefs will combine the beliefs of both graphs by selecting the beliefs of the higher, more accurate graph. These final combined beliefs will also be normalized and is accounted for in the BP-TwoGraphs algorithm. In the STUMP algorithm, these final beliefs at each node correspond to probabilities for each disparity at each pixel vertex. STUMP chooses to select the best disparity (highest probability) for each node (pixel). Details of this process will be explained in the following chapters. In conclusion, there is a general description of belief propagation followed by a short example below explaining of how beliefs are propagated within BP-TwoGraphs.

Considering a Markov random field with n vertices, the joint probability that each vertex i is in the state x_i is defined by equation (2.5.1) where Z is the normalization constant and y_i is the observed value at that node. To determine the most likely arrangement of states one could calculate the joint probability of all combinations of states by means of equation (2.5.1) although this would take exponential time in the number of vertices to complete.

$$p(x_1,\ldots,x_n) = \frac{1}{Z}\prod_{i=1}^{n}\phi_i(x_i,y_i)\prod_{(ij)}\psi_{ij}(x_i,x_j) \qquad (2.5.1)$$

When computing the joint probability for all combinations of states it is known that many of the terms are identical for many combinations. As a result belief propagation takes advantage of this property to calculate the joint probability efficiently. Considering a system of beliefs and messages one can say that the vertex k is in state x_k is defined by the following equation (2.5.2). The term z_k normalizes beliefs so that $\sum_{x_k \in X_k} b_k(x_k) = 1$ and $N(k)$ is the set of the neighbors of vertex k. The term $m_{jk}(x_k)$ is the message that vertex j passes to vertex k and is defined by equation (2.5.3) where $N(j) \backslash k$ is the set of the neighbors of vertex j, excluding vertex k. This message indicates how likely vertex j thinks it is, given the knowledge accumulated from its ancestors, that vertex k is in state x_k. These messages are responsible for storing the common terms between joint probabilities for efficient belief propagation.

$$b_k(x_k) = \frac{1}{z_k} \phi_k(x_k, y_k) \prod_{j \in N(k)} m_{jk}(x_k) \tag{2.5.2}$$

$$m_{jk}(x_k) = \sum_{x_j \in X_j} \phi_j(x_j, y_j) \psi_{jk}(x_j, x_k) \prod_{i \in N(k) \backslash k} m_{ij}(x_j) \tag{2.5.3}$$

We will demonstrate a simple example of computing a belief. Consider the graph segment shown in Figure 2.5.3 below. There are three nodes $(1,2,3)$ with node 2 being the pixel vertex of interest in which we wish to find the belief.

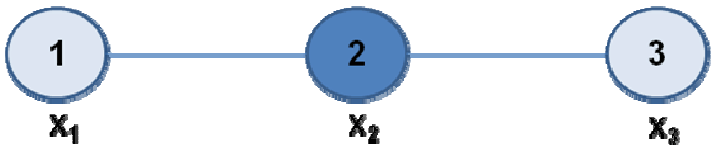


Figure 2.5.3: Example of Computing a Belief. The graph segment above shows 3 nodes with respective states (x_1, x_2, x_3). The vertex of interest is node 2 (blue). The ψ function between nodes 1 and 2 is $\psi_{12}(x_1, x_2)$ and the ψ function between nodes 3 and 2 is $\psi_{32}(x_3, x_2)$. Below we will explain how the belief that node 2 is in state x_2 is computed.

In order to determine the most likely arrangement of states one could calculate the joint probability for all combinations of states although this would take exponential time in the number of vertices to complete so instead we calculate the belief. This approach of solving the joint probability for $p(x_1, x_2, x_3)$ is given by the following equation (2.5.4).

$$p(x_1, x_2, x_3) = \phi_1(x_1, y_1)\phi_2(x_2, y_2)\phi_3(x_3)\psi_{12}(x_1, x_2)\psi_{32}(x_3, x_2) \qquad (2.5.4)$$

This equation (2.5.4) can be rewritten as the following equation (2.5.5) which we will show that this equation (2.5.5) will be equivalent of computing the belief of vertex 2 at state x_2.

$$p(x_2) = \sum_{x_1 \in X_1, x_3 \in X_3} p(x_1, x_2, x_3) \qquad (2.5.5)$$

The belief that vertex 2 is in state x_2 is defined by equation (2.5.6) with messages from vertex 1 to vertex 2 and messages from vertex 3 to vertex 2 defined in equations (2.5.7, 2.5.8) respectively, which are responsible for storing the common terms between joint probabilities to make belief propagation efficient.

$$b_2(x_2) = \phi(x_2, y_2) m_{12}(x_2) m_{32}(x_2) \qquad (2.5.6)$$

$$m_{12}(x_2) = \sum_{x_1 \in X_1} \phi(x_1, y_1) \psi_{12}(x_1, x_2) \qquad (2.5.7)$$

$$m_{32}(x_2) = \sum_{x_3 \in X_3} \phi(x_3, y_3) \psi_{32}(x_3, x_2) \qquad (2.5.8)$$

Given equations (2.5.7, 2.5.8), equation (2.5.6) can be rewritten as the following equation (2.5.9). This equation shows that the product of the terms ϕ and ψ is exactly the joint probability $p(x_1, x_2, x_3)$.

$$b_2(x_2) = \sum_{x_1 \in X_1} \sum_{x_3 \in X_3} \phi_1(x_1, y_1) \phi_2(x_2, y_2) \phi_3(x_3, y_3) \psi_{12}(x_1, x_2) \psi_{32}(x_3, x_2) \qquad (2.5.9)$$

As a final result of equations (2.5.4, 2.5.5, 2.5.9) we can conclude that $b_2(x_2) = p_2(x_2)$. Calculating beliefs rather than joint probabilities yields a great improvement in complexity presented by Kai Ju Liu [17]. Following the formulaic example of propagating beliefs, we will now show how messages propagate within the horizontal and vertical graphs. Figure 2.5.4 as displayed below, shows how messages within the horizontal graphs propagate to node 8.

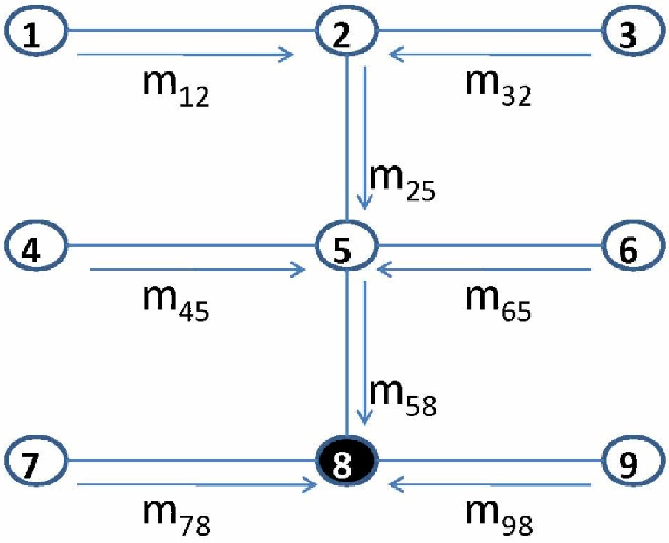


Figure 2.5.4: Horizontal Graph Message Propagation. The belief at node 8 (black node), is calculated via the following messages $\{m_{78}(x_8), m_{58}(x_8), m_{98}(x_8)\}$, satisfying the equation $b_8(x_8) = \Phi_8(x_8) m_{78}(x_8) m_{98}(x_8) m_{58}(x_8)$ where $m_{58}(x_8)$ is assumed to have already been calculated.

The following Figure 2.5.5 below shows how messages are propagated within the vertical graph for the node 8. Between both the horizontal and vertical graphs, values can be reused. Within the horizontal graph, the horizontal messages do not need to be recalculated. The same is true for the vertical graph in that the vertical messages also do not need to be recalculated. However, messages in the horizontal graph cannot be reused within the vertical graph and vice versa.

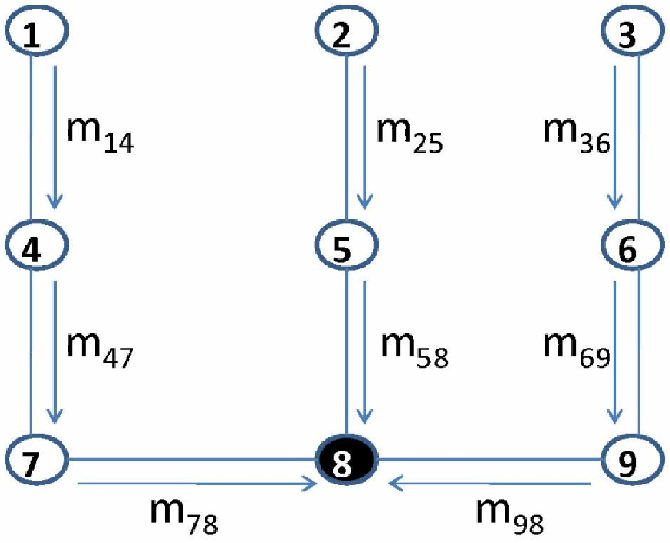


Figure 2.5.5: Vertical Graph Message Propagation. The belief at node 8 (black node), is calculated via the following messages $\{m_{78}(x_8), m_{58}(x_8), m_{98}(x_8)\}$, satisfying the equation $b_8(x_8) = \Phi_8(x_8) m_{78}(x_8) m_{98}(x_8) m_{58}(x_8)$ where the messages $\{m_{78}(x_8), m_{58}(x_8), m_{98}(x_8)\}$ are assumed to have already been calculated.

Concluding our explanation of belief propagation under the BP-TwoGraphs algorithm, we will mention briefly the complexity as stated by Kai Lu Liu [17]. Assuming that there are N nodes in the Markov random field (meaning N is the number of pixels of the stereo pair) and S is the number of states (meaning that S is the range of disparity), then for the BP-TwoGraphs algorithm there are $12NS^2 + 8NS$ multiplications, $8NS^2$ additions, and $2NS$ stores. These numbers will be introduced once again as we discuss the final complexity of the STUMP algorithm in chapter 4.

3 Stereo Correspondence

We will now introduce our contribution to the stereo correspondence problem. First we will define the matching space which will state which pixels can correspond to each other in the stereo pair of images. Next we will model the problem in the cyclopean coordinate system indicating the advantages and intuition for it. Before we introduce our algorithm we will first explain how we extract the data from our images (features). Then we will explain the constraints used and how we model our surfaces. Afterwards we will explain how we incorporate belief propagation into our correspondence algorithm and finally we will discuss a multi-scale approach. Following our theory will be a series of experiments and analysis.

3.1 The Matching Space

The matching space of the stereo correspondence algorithm can be defined as the space containing the pairs of pixels belonging to the epipolar lines between the left and right images. A point in the matching space represents the match of a pixel in the left image with a pixel in the right image. Consider the following figure 3.1.1, which depicts points on a surface (the shaded region) projecting through the stereo pairs of

cameras onto the left and right epipolar lines of the image planes. Some of the points on the surface can be seen by both cameras, (points F, D, C, A) while other points can only be seen by either the left camera (point B) or right camera (point E). Depth discontinuities and strongly tilted surfaces can yield the same images with half occluded pixels. One may also notice in the figure below that due to pixel discretization, points A and C in the right image plane are neighbors.

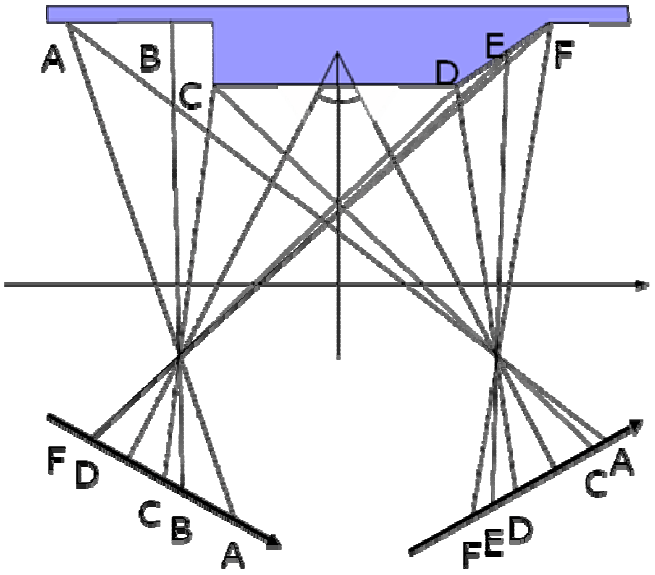


Figure 3.1.1: Matching Space. The above diagram depicts the points of each epipolar line projecting onto a surface through each camera. Some points on the surface can be seen by both cameras while other points can only be seen through either the left or the right camera.

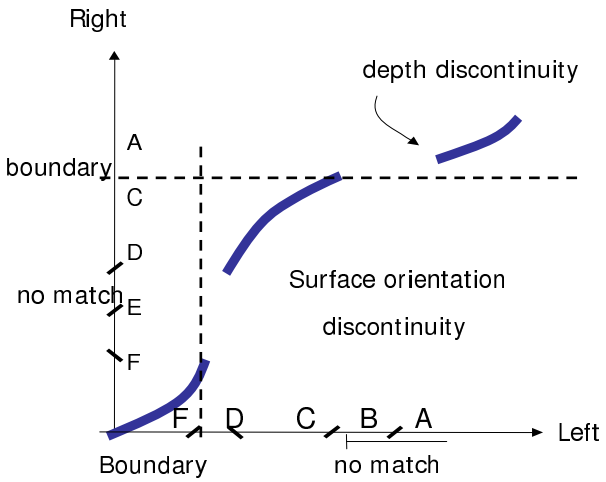


Figure 3.1.2: Matching Space Graph. This graph depicts the matching space as described by Figure 3.1.1. The blue segments show the pixel matches in the left and right image planes while the breaks in the blue curve show the occluded and tilted surfaces which indicate a change in depth or disparity.

The matching space can be thought of as a graph, G(V,E), where nodes or vertices describe the matching of pixels. As we will address later in the following section, the graph also includes the match of sub-pixels, which are the locations halfway between pixels. Edges in this graph represent the link between nodes, and in particular neighborhood relations (edges) in the graph will be examined.

3.2 Cyclopean Coordinate System

In stereo vision it is perhaps more intuitive to consider a cyclopean coordinate system, rather than two separate coordinate systems for the left and right image planes independently. Bela Julesz [15] mentioned that the stereo correspondence is determined within a single view: the cyclopean view, which is derived by fusing both the left and right views of a stereo pair. Given the left and right coordinate systems, defined by coordinates (e,l) and (e,r), we can derive the cyclopean coordinate system with coordinates e,x,w as shown in Figure 3.2.1. The coordinate e is the epipolar line which is unchanged under the transformation from left and right into cyclopean coordinate systems. The coordinates l and r are the pixels along each epipolar line e for the left and right coordinate systems respectively. The linear transformation (45 degree rotation) can be described as follows. In order to change from the left and right coordinate systems into the cyclopean coordinate system, the equations (3.2.1a, 3.2.1b) are used. Similarly, in order to change from the cyclopean coordinate system into the left and right coordinate systems, equations (3.2.2a, 3.2.2b) are used.

$$x = \frac{r+l}{\sqrt{2}} \qquad (3.2.1a) \qquad\qquad w = \frac{r-l}{\sqrt{2}} \qquad (3.2.1b)$$

$$r = \frac{x+w}{\sqrt{2}} \qquad (3.2.2a) \qquad\qquad l = \frac{x-w}{\sqrt{2}} \qquad (3.2.2b)$$

Alternatively the above equations (3.2.1a, 3.2.1b and 3.2.2a, 3.2.2b) respectively can be written as equations (3.2.3 and 3.2.4).

$$\begin{pmatrix} x \\ w \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} r \\ l \end{pmatrix} \qquad (3.2.3)$$

$$\begin{pmatrix} r \\ l \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x \\ w \end{pmatrix} \qquad (3.2.4)$$

The triplet (e, x, w) represents a node in the graph of the cyclopean coordinate system. This coordinate system is interpreted as follows. Instead of representing a match by the two image points on the left and right images, the cyclopean eye represents a match by a coordinate x and a disparity w. Since the disparity w is mapped to depth, one may say that the coordinate system represents a match by its 3D representation, meaning a generalized coordinate x and e (replacing the left and right eyes) and the corresponding depth. It is worth noting that we do have more x values than r or l values, as l varies from 0 to N and r varies from 0 to N while x varies from 0 to $\sqrt{2}N$. The cyclopean eye will see the 3D points that perhaps one eye may not see due to an occlusion. More precisely, a point on the left eye that does not correspond to any point in the right eye such as a half occluded point, may land on some x coordinate and disparity w. Analogously, the right eye occluded points will have an x and w coordinate. Consider the following Figure 3.2.1.

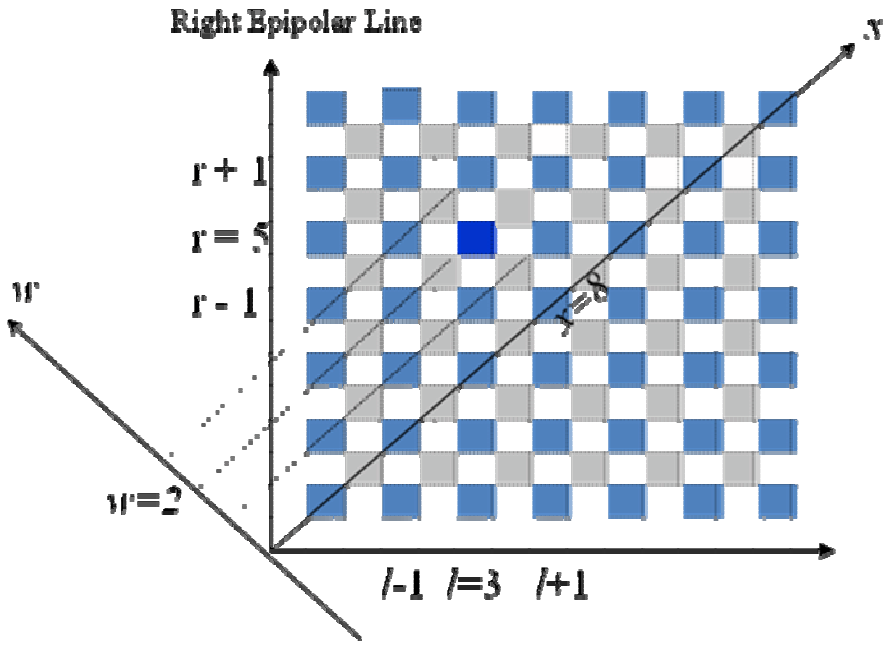


Figure 3.2.1: Cyclopean Eye. Cyclopean coordinate system defined as (x,w) for each epipolar line. x is the single eye coordinate system and w is the disparity axis. Since we are only restricted to integer values for l and r, not every (x,w) pair has a correspondence to l and r. For $(x+w)$ *even* we have integer values for l and r. For $(x+w)$ *odd* we have sub-pixel locations for l and r. This figure depicts the 45 degree rotation of (l,r) and (x,w).

The above figure shows the transformation of left and right pixel (l,r) coordinates mapped into cyclopean coordinates (x axis) along with a disparity (w axis). This linear transformation of coordinates is shown by equations (3.2.3, 3.2.4) above. In the diagram above (Figure 3.2.1), the dark blue square (location of interest) shows the transformation between coordinate systems. Assuming that the blue square is at location $l=3$ and $r=5$ in the (l,r) coordinate system, then by

equations (3.2.1a, 3.2.1b), the blue square is at $x = 8$ and $w = 2$ in the (x, w) cyclopean coordinate system. Since l and r are pixel coordinates, we are restricted to integer values. Therefore $l, r = 0, \ldots, N-1$ and $x = 0, \ldots, 2N-2$ and $w = -N+1, \ldots, 0, \ldots, N-1$. When $(x+w)$ is even, pixels l and r are integer values. However, when $(x+w)$ is odd, we have sub-pixel locations. Therefore the cyclopean coordinate system for integer values of (x, w) includes a sub-pixel image resolution.

The cyclopean coordinate system therefore represents disparity values w with sub-pixel accuracy, meaning that w corresponds to pixel matching and sub-pixel matching from the left and right images.

3.3 Features

The question of matching pixels in a stereo pair of images can be thought of as matching the features of the scene portrayed in the stereo pair. Each pixel contains data, and features are the data which is extracted from these pixels. Currently we use grayscale source images; therefore the data of the pixels (features) are derived from grayscale values.

A feature in the left image can match a feature in the right image and a score function for the matching can then be assigned. In the cyclopean coordinate system,

the score function will be assigned to a coordinate (e, x, w). Next we will investigate the use of features based on intensity values (grayscale) averaged over some scale s and orientation θ. These features are referred to as $\hat{I}^L(e,l,\theta,s)$ and $\hat{I}^R(e,r,\theta,s)$ as is illustrated in Figure 3.3.1 below.

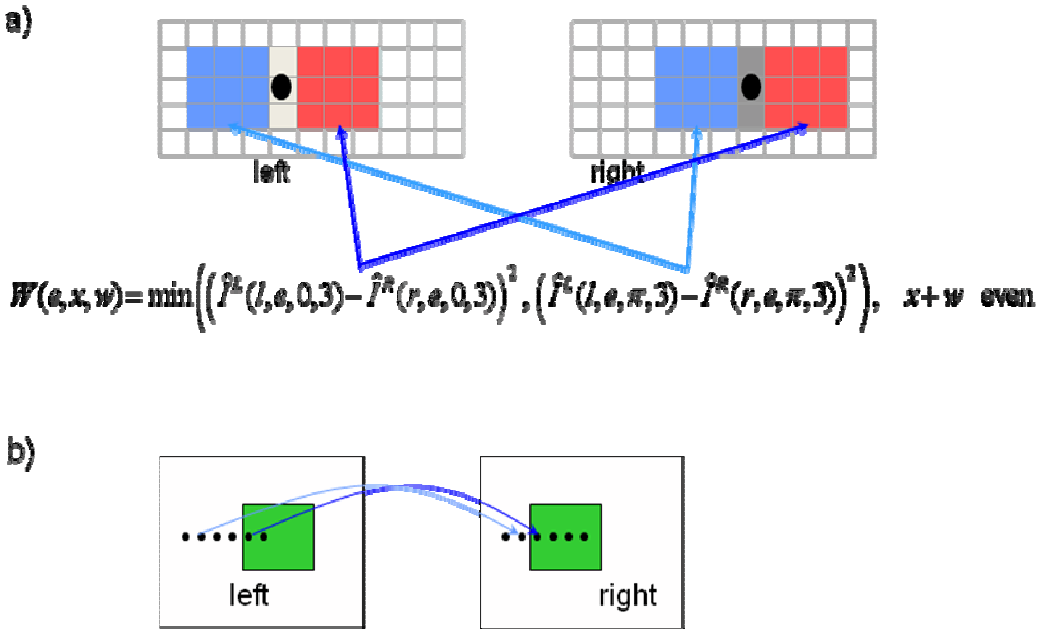


Figure 3.3.1: Window Matching. Two left eye windows, with $\theta=0$ and $\theta=\pi$, are considered (red and blue) to match two corresponding right eye windows (red and blue). The averaged value of each window is actually used to establish the match. We also consider matching windows with $\theta=\pi/2$ and $\theta=3\pi/2$. Other scoring functions could be considered. In this example the scale is set to 3.

In the Figure 3.3.1 above, each pixel of both the left and right images contains a window of size scale s for a given orientation θ. The images (a) show a magnified

view of the green square stereo pair (b). In actuality the images (a) can be of any pixel of the stereo pair (b), not just the edge of the square.

The precise formula for $\hat{I}(x, y, \theta, s)$, for $\theta = 0, \pi/2, \pi, 3\pi/2$ is given by

$$\hat{I}(\vec{x}, \theta, s) = \frac{1}{4}\left(\tilde{I}\left(\vec{x}, \theta, s\right) + \tilde{I}\left(\vec{x} + R(\theta)\vec{e}_y, \theta, s\right) + \tilde{I}\left(\vec{x} - R(\theta)\vec{e}_y, \theta, s\right) + \tilde{I}\left(\vec{x} + R(\theta)\vec{e}_x, \theta, s\right)\right) =$$

$$\frac{1}{4}\begin{pmatrix} \tilde{I}(x, y, \theta, s) + \tilde{I}(x - \sin\theta, y + \cos\theta, \theta, s) + \\ \tilde{I}(x + \sin\theta, y - \cos\theta, \theta, s) + \tilde{I}(x + \cos\theta, y + \sin\theta, \theta, s) \end{pmatrix} \qquad (3.3.1)$$

where $\vec{x} = \begin{pmatrix} x \\ y \end{pmatrix}$, $\vec{e}_x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\vec{e}_y \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$,

$$\tilde{I}(x, y, \theta, s) = \frac{1}{s}\sum_{i=0}^{s-1} I(x + i\cos\theta, y + i\sin\theta)$$

In the equation above $I(i, j)$ is defined as the intensity of pixel (i, j). The scale s measures the window size in which the average is to be taken. Note that (x, y) is to be replaced by (e, l) or (e, r) accordingly if one wants to compute $\hat{I}^L(e, l, \theta, s)$ or $\hat{I}^R(e, r, \theta, s)$.

We want to create a score function and following Geiger, Ladendorf and Yuille [8] we take into consideration that near half occlusions on one side of the window will have a better match than the side that is not half occluded. We have considered the following window matching formula:

$$W_e(e,x,w,s) = W_e(e,l,r,s) =$$
$$\min\left(\left(\hat{I}^L(l,e,\theta,s) - \hat{I}^R(r,e,\theta,s)\right)^2, \left(\hat{I}^L(l,e,\theta+\pi,s) - \hat{I}^R(r,e,\theta+\pi,s)\right)^2\right) \qquad (3.3.2)$$

When $(x+w)$ is odd, the coordinates l and r are half integers, therefore an interpolated average value for the intensity values needs to be computed. For half integer values of l and r we use the following formulas.

$$\hat{I}^L(e,l,\theta,s) = \frac{1}{2}\left(\hat{I}^L(e,\lfloor l \rfloor + 1,\theta,s) + \hat{I}^L(e,\lfloor l \rfloor,\theta,s)\right) \qquad (3.3.3)$$

$$\hat{I}^R(e,r,\theta,s) = \frac{1}{2}\left(\hat{I}^R(e,\lfloor r \rfloor + 1,\theta,s) + \hat{I}^R(e,\lfloor r \rfloor,\theta,s)\right) \qquad (3.3.4)$$

where $\lfloor x \rfloor$ is the floor value of x. Our proposed score function is

$$CostPixel(e,x,w,s) = 2 + \sqrt{\frac{W_{\theta=0}(e,x,w,s)}{T}} + \sqrt{\frac{W_{\theta=\pi/2}(e,x,w,s)}{T}} \qquad (3.3.5)$$

where T is a parameter to be estimated. So in order to evaluate the quality of a match we are matching horizontal window features $\theta = (0,\pi)$ and vertical window features $\theta = (\pi/2,\ 3\pi/2)$. If the intensities match perfectly then the window scores are zero and the cost is 2 which is the minimum cost. We also assign a score function for the match of intensity differences which are edges. We propose the following formula

$$CostEdge(e,x,w,s) = \sqrt{\frac{D^-_{\theta=0}(e,x,w,\theta,s)+1}{D^+_{\theta=0}(e,x,w,\theta,s)+1}} + \sqrt{\frac{D^-_{\theta=\pi/2}(e,x,w,\theta,s)+1}{D^+_{\theta=\pi/2}(e,x,w,\theta,s)+1}} \qquad (3.3.6)$$

where the intensity differences in the left and right eyes are defined as follows

$$D\hat{I}^L(e,l,\theta,s) = \hat{I}^L(e,l,\theta,s) - \hat{I}^L(e,l,\theta+\pi,s) \qquad (3.3.7)$$

$$D\hat{I}^R(e,r,\theta,s) = \hat{I}^R(e,r,\theta,s) - \hat{I}^R(e,r,\theta+\pi,s) \qquad (3.3.8)$$

and matching differences as described in the cyclopean coordinates, requires the terms

$$\left|D\hat{I}_\theta^+(e,x,w,s)\right| = \left|D\hat{I}^R(e,r,\theta,s) + D\hat{I}^L(e,l,\theta,s)\right| \qquad (3.3.9)$$

$$\left|D\hat{I}_\theta^-(e,x,w,s)\right| = \left|D\hat{I}^R(e,r,\theta,s) - D\hat{I}^L(e,l,\theta,s)\right| \qquad (3.3.10)$$

The quantity $\left|D_\theta I^+(e,x,w,s)\right|$ captures how much magnitude of the left and right intensity differences there is for a given angle direction, if they are the same sign. If they are of opposite signs, it yields very small values. The quantity $\left|D_\theta I^+(e,x,w,s)\right|$ shows how similar the two derivatives of left and right images are. The similar, the smaller is the value. Note that for homogenous regions, both the *CostPixel* and *CostEdge* gives an equal score of 2. In this scenario the intensities match very well and the derivatives are near zero value, therefore the matching between them is also good. Moreover, well defined intensity edges will yield even lower *CostEdge* values, and therefore will be the best features to match. Our total score function is then

$$\phi(e,x,w,s) = CostPixel(e,x,w,s) + CostEdge(e,x,w,s) \qquad (3.3.11)$$

which leads to a probability model:

$$P(I^L, I^R \mid e,x,w) = \frac{1}{Z} e^{-\phi(e,x,w,s)} \qquad (3.3.12)$$

where $\phi(e,x,w,s) =$

$$2+\sqrt{\frac{W_{\theta=0}(e,x,w,s)}{T}}+\sqrt{\frac{W_{\theta=\pi/2}(e,x,w,s)}{T}}+$$
$$\sqrt{\frac{D^{-}_{\theta=0}(e,x,w,\theta,s)+1}{D^{+}_{\theta=0}(e,x,w,\theta,s)+1}}+\sqrt{\frac{D^{-}_{\theta=\pi/2}(e,x,w,\theta,s)+1}{D^{+}_{\theta=\pi/2}(e,x,w,\theta,s)+1}} \qquad (3.3.13)$$

Occlusions are regions in which feature matching does not occur. In order to consider occlusions we introduce an occlusion field $O(e,x,w)$, which is 1 if an occlusion occurs at location (e,x,w) and zero otherwise. Thus the likelihood of left and right images given the feature matching must take into account the occurrence of an occlusion. We modify Equation (3.3.12) to include occlusions as

$$P(I^L, I^R \mid e,x,w) = \frac{1}{Z} e^{-(1-O(e,x,w))\,\phi(e,x,w,s)} \qquad (3.3.14)$$

where $O(e,x,w) = 0,1$

3.3.1 Multi-Scale Feature Integration and Entropy

The feature matching thus far, $\phi(e,x,w,s)$, varies at different scales. Ishikawa [14] has shown a method of integrating features from different scales. He combines different features (such as features at different scales) using a convex linear combination of the type $\Phi(e,x,w) = \sum_{s=1}^{S} a_s(e,x)\phi(e,x,w,s)$ (3.3.15)

with $a_s(e,x) \geq 0, 1 = \sum_{s=1}^{S} a_s(e,x)$, where the main idea is to look for a given feature at one location (e,x) and a positive coefficient $a_s(e,x)$ that is chosen according to the feature discrimination power across different disparity values w. We do know that features are being used to discriminate which disparity w is best assigned to each location (e,x). A measure of discrimination power can be described as follows. Each feature response assigns, via Equation (3.3.12), a probability of preferring a disparity w as $P_{e,x,s}(w) = \frac{1}{Z} e^{-\phi(e,x,w,s)}$ (3.3.16)

where $Z(e,x,w) = \sum_{w=-D}^{D} e^{-\phi(e,x,w,s)}$. This probability has entropy H associated to it defined as $H_s(e,x) = -\sum_{w=-D}^{D} P_{e,x,s}(w) \log P_{e,x,s}(w) = \log Z(e,x,s) + \phi(e,x,s)$ (3.3.17).

The smaller the entropy is, the greater the discrimination. As a result, the distribution is peaked at the best (smaller) matches $\phi(e,x,w,s)$. We can consider a quantity that is a type of complement of the entropy $\tilde{H}_s(e,x) = H_{e,x} - H_s(e,x)$ (3.3.18)

where $H_{e,x} = \sum_{s=1}^{S} H_s(e,x)$. Therefore the larger $\tilde{H}_s(e,x)$ is, the more discrimination found. We can then simply use the coefficients $a_s(e,x)$ to be proportional to $\tilde{H}_s(e,x)$ and normalize it. More precisely,

$$a_s(e,x) = \frac{1}{(s-1)} \frac{\tilde{H}_s(e,x)}{H_{e,x}} \quad (3.3.19)$$

In this way we can construct the feature match that is scale

invariant: $\Phi(e,x,w) = \sum_{s=1}^{S} a_s(e,x)\phi(e,x,w,s)$ (3.3.20)

and we also obtain the data probability that a three dimensional surface coordinate (e,x,w) will produce a feature match $\Phi(e,x,w)$ including modeling occlusions

as $P(I^L, I^R \mid \{O,w\}) = \frac{1}{Z_I} \prod_{e=1}^{N} \prod_{x=1}^{2N} e^{-(1-O(e,x,w))\,\Phi(e,x,w)}$ (3.3.21)

We now focus on the prior knowledge that we have regarding the structure of the surfaces that appear in the real world.

3.4 Uniqueness – Opaque Constraint

We define our uniqueness – opaque constraint to state that there should only be one disparity value (a single depth value) associated to each cyclopean coordinate (e,x) as is shown in the following Figure 3.4.1. There is an underlying assumption that objects are opaque and so a 3D point P can be seen by the cyclopean coordinate (e,x) and disparity w which will not allow all other disparity values (to be seen). Points that are closer than P along the same (e,x) coordinate must be transparent air, and further away points will not be seen since P is already seen and is opaque. However, multiple matches for the left eye points or the right eye points are allowed. This is indeed required to model not only tilted surfaces but also occluded surfaces as

we will later address. This constraint is physically motivated and can be easily understood in the cyclopean coordinate system.

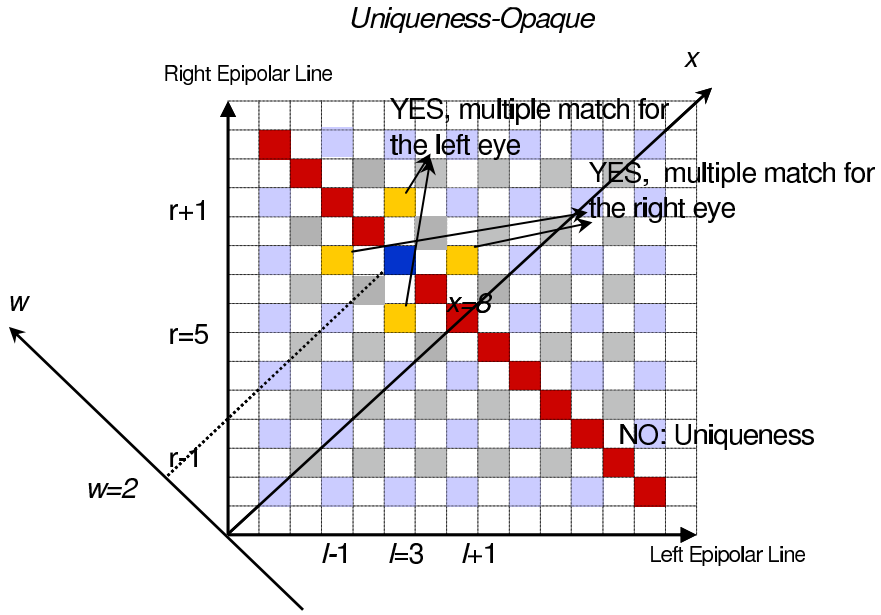


Figure 3.4.1: Uniqueness-Opaque Constraint. Given that the $l = 3$ and $r = 5$ pixels are matched (blue square), then the red squares represent violations of the uniqueness-opaqueness constraint, while the yellow squares represent unique matches in the cyclopean coordinate system, but multiple matches in the left or right eye coordinate system.

3.5 Smooth Surface Constraint

In nature most surfaces are smooth in depth compared to their distance to the observer, yet depth discontinuities also occur. Many authors claim that smoothness implies an ordering constraint, where points to the right of $\vec{q}_l$ (defined in equation 2.1.17) cannot match points to the left of $\vec{q}_r$ as shown in Figure 3.5.1 below.

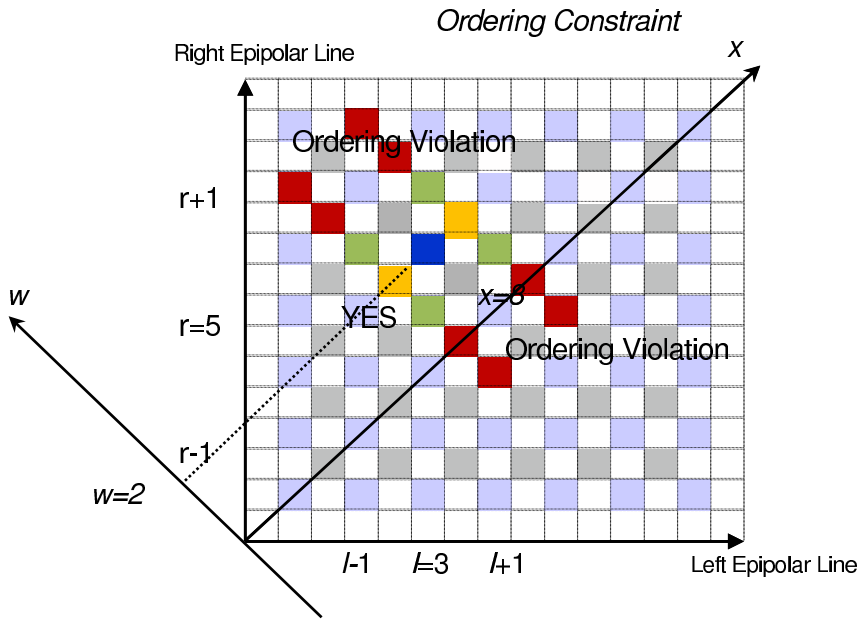


Figure 3.5.1: Ordering Constraint. Given that the $l=3$ and $r=5$ pixels are matched (blue square), then the red squares represent violations of the ordering constraint while the yellow and green squares represent smooth matches. The yellow (flat) and green (tilted or occluded) squares represent the type of surface as will be discussed next.

Note that in order to account for the tilted surfaces solution, the order constraint must accept points to the right of $\vec{q}_l$ to also match $\vec{q}_r$ and points to the right of $\vec{q}_r$ to also match $\vec{q}_l$ as shown in Figure 3.4.1. Further examination of a configuration with occlusions will show that discontinuities in disparity will result in half-occlusions, meaning pixels in one eye that do not have any correspondence (see Figure 3.5.2). Moreover, in these half occlusion cases, some pixels from one eye will not be matched to any pixel in the other eye ("$l+1$" in Figure 3.5.2). Other pixels in one eye will have multiple matches in the other eye ("$r-1$" in Figure 3.5.2). In fact, the number of pixels unmatched in the left image is the same as the number of multiple matches in the right image and vice-versa. In the cyclopean eye, this is described as discontinuities in the disparity field $w(e,x)$.

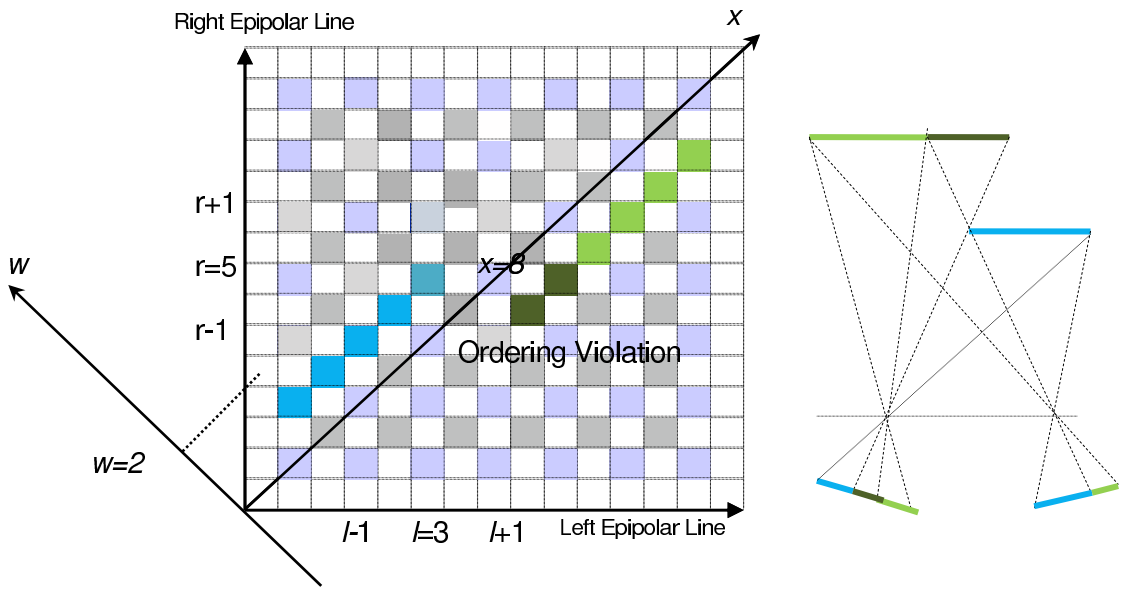


Figure 3.5.2: A Valid Surface and a Violation of the Ordering Constraint. While our approach adopts the ordering constraint, for simplicity of computations, it can be extended to cases such as in this figure where the ordering constraint is violated. Note that in these cases, some pixels will not be matched to any pixel, e.g. "$l+1$" and other pixels will have multiple matches, e.g. "$r-1$". In fact, the number of pixels unmatched in the left image is the same as the number of multiple matches in the right image.

If we restrict the changes of the disparity field to be limited to one unit for neighbor coordinates x, then our model is a combination of the ordering constraint and the uniqueness constraint, which imposes a disparity gradient constraint. More precisely, from $(e, x-1)$ to (e, x), the disparity $w(e, x)$ cannot change in magnitude by more than 1, meaning $|w(e, x) - w(e, x-1)| \leq 1$. In our work, we will utilize this

constraint and examine in detail the possible scenarios in 3D corresponding to the disparity gradient constraint so as to formulate our model of stereo.

3.5.1 Occluded, Tilted, and Flat Surfaces

Why do we need to further analyze a prior model for surfaces beyond simply imposing a smooth constraint? Why do we need to investigate the differences between flat surfaces, tilted surfaces, and occluded surfaces? These surfaces compete among themselves to be the optimal solutions, and against any arbitrary (and more complex) surfaces. Here are some examples to illustrate the need for modeling these surfaces.

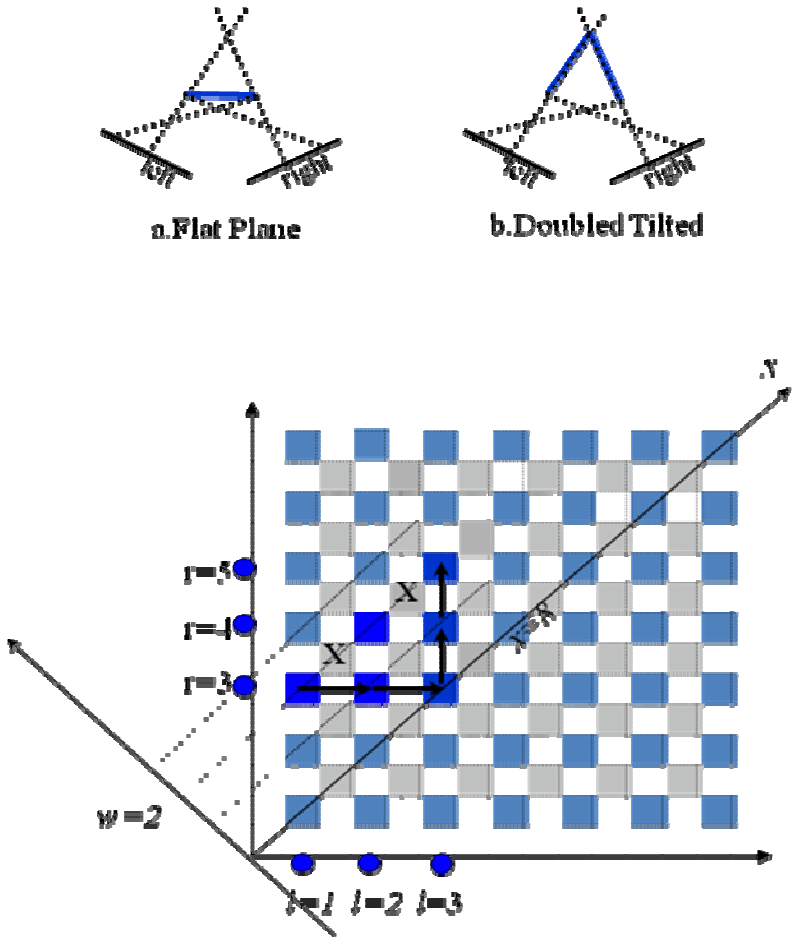


Figure 3.5.3: Flat Plane versus Double Tilted Plane. The data and the score of matching data in (a) and (b) are the same. Since we are not considering curvature preferences, the preference for flat surfaces must be built as a prior model of surfaces. The geometrical views (a) and (b) are shown in the cyclopean view of the graph via the X path for the flat plane solution and via the arrow path for the double tilted solution.

In order to examine occluded surfaces and to compare them with tilted or occluded surfaces we must first choose a representation for occluded solution in the

matching space. A node in the matching space represents either a match between pixels or a match between sub-pixels. An occluded surface will be represented as a path of varying disparity values through the sub-pixel matching nodes, while competing tilted surfaces will be represented by a path of varying disparity values through the pixel matching nodes. This choice is arbitrary. A flat surface, which is a surface of constant disparity, will be a path through both sub-pixel and pixel matching nodes at constant disparity. This representation is depicted in the following Figure 3.5.4.

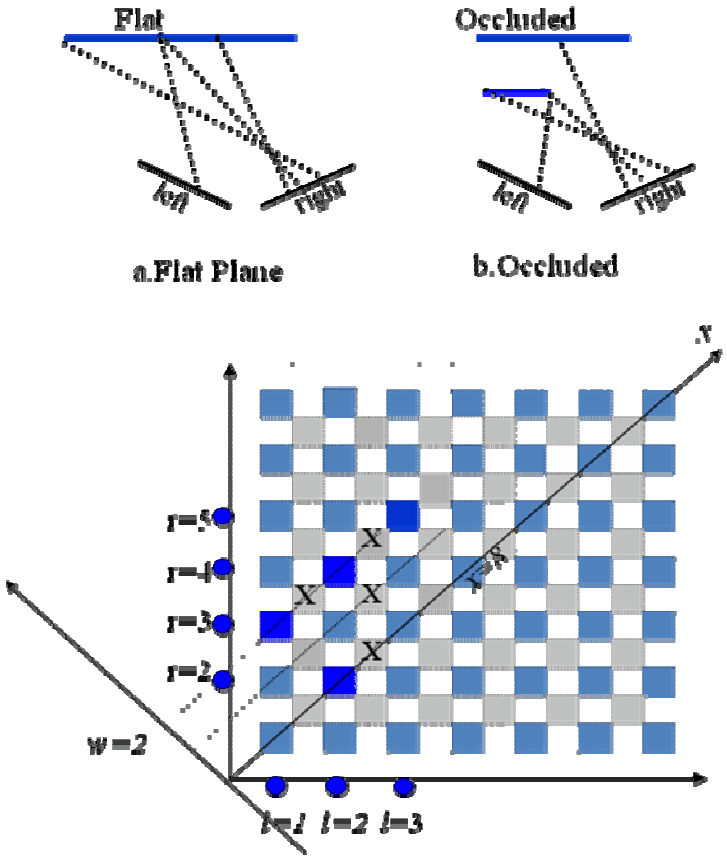


Figure 3.5.4: Flat Plane versus Occluded Plane. The data and the score of matching the data in (a) and (b) are the same, if we simply set $O(e,x)=0$ at Equation (3.3.21). Thus, modeling the occlusion and introducing some penalty for setting $O(e,x)=1$ is necessary (otherwise $O(e,x)=1$ will always be the lowest cost). Some cost for occlusion should be added to bias the solution towards a flat surface. Moreover, occlusions are really not described by a match of features, but rather by a lack of a match of features. Thus, a better modeling of occluded solutions must be considered.

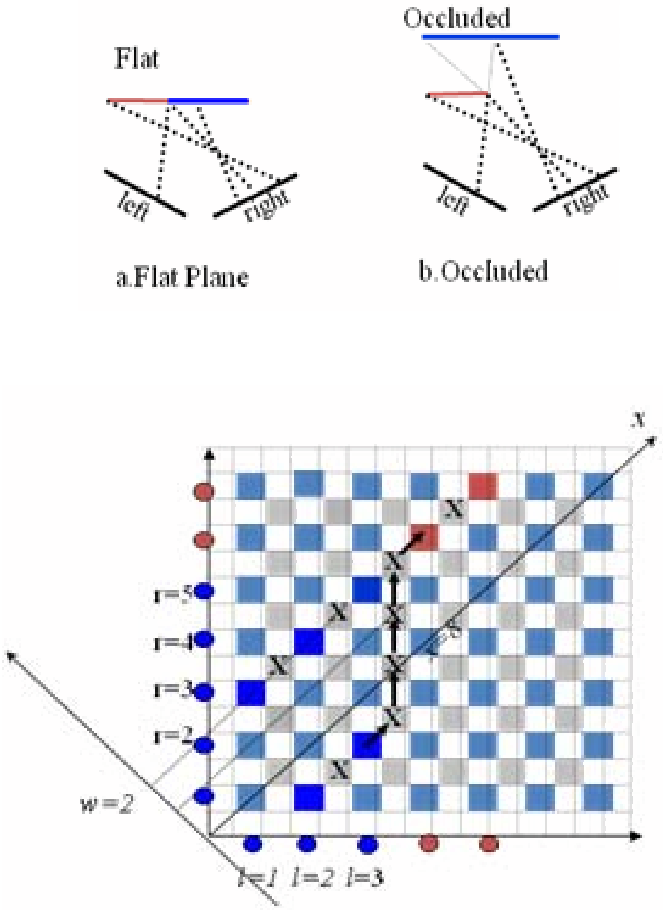


Figure 3.5.5: Flat Plane versus Occluded Plane (Color). The occlusion solution in this case, should be slightly better than the solution in the previous case in Figure 3.5.4. With the red color replaced by blue, there exists a contrast which improves the chances of a discontinuity in depth.

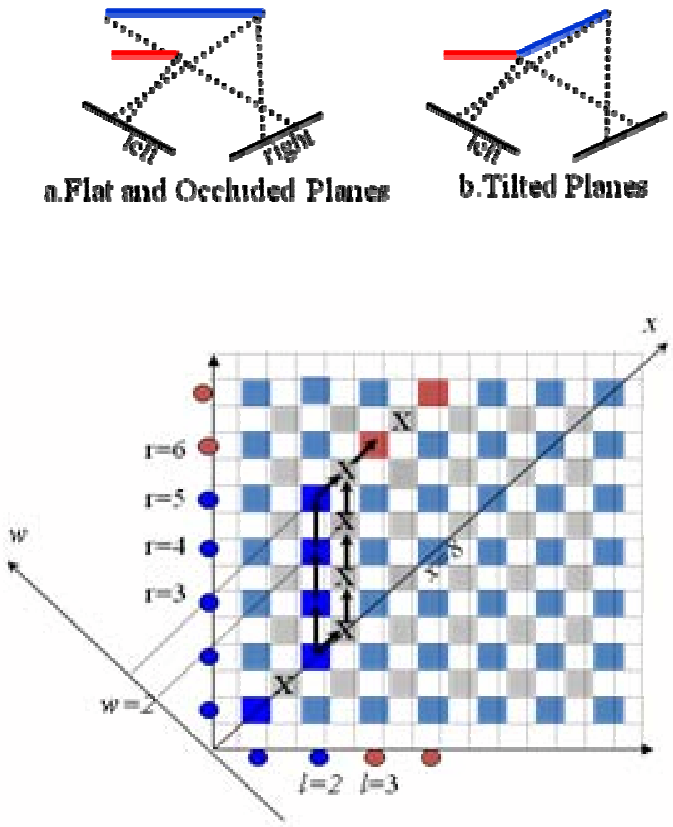


Figure 3.5.6: Flat and Occluded Planes versus Tilted Planes. We do expect these solutions to be close, although the tilted plane may be slightly more likely to occur by human perception. Again, the occlusion solution should not reflect any matching between features that are not matched.

3.6 Modeling Surfaces

Taking into consideration the constraints mentioned above, we will now proceed to model surfaces. It is worth noting that all constraints mentioned above are

constraints on the tangent plane near a matching point. Essentially there is a bias for the tangent planes to be flat, although they may be tilted or discontinuous (occluded).

The flat surface bias and the tilted surface bias can be described within the smoothness constraint by imposing different criteria for the transitions of the disparity values. It is also possible to model occlusions in this fashion. As we will show, it is possible to insure that all these surfaces can be modeled by specifying the transition probabilities of the disparities that satisfy the disparity gradient $\text{limit}\left|w(x)-w(x-1)\leq 1\right|$.

We are modeling the occlusion path through the nodes of varying disparity representing sub-pixel matches, which are described by $(x+w)$ odd values. We can then constrain this solution by setting $O(e,x,w)=0$ for any location where $(x+w)$ is even, so an occlusion cannot occur at pixel matches. Tilt is represented by a path through the nodes of varying disparity representing pixel matches, which are described by $(x+w)$ *even* values. Moreover, since the disparity changes are limited to one unit, we can model both tilt and occlusion with the following probability model.

$$P(\{O(e,x,w),w(x,e)\})=$$
$$\prod_{e=0}^{N-1}\prod_{x=0}^{2N-1}\left[\left(1-\theta\left(\mathrm{mod}(x+w,2)\right)\right)P_{tilt}(w,w')+\theta(\mathrm{mod}(x+w,2))P_{occl}(O,w,w')\right] \quad (3.6.1)$$

$$\theta(y)=\begin{cases}1 & \text{if } y>0\\ 0 & \text{otherwise}\end{cases}$$

where $$\mathrm{mod}(x+w,2)=\begin{cases}1 & \text{if x+w odd}\\ 0 & \text{if x+w even}\end{cases}$$

$$O=0,1,\ \ w'=w-1,w,w+1$$

Combining this prior model with the data probability, we obtain the posterior model as

$$P(\{w(x,e),O(e,x,w)\}\mid I^L,I^R)=$$
$$\frac{1}{Z}\prod_{e=0}^{N-1}\prod_{x=0}^{2N-1}\left[\begin{array}{l}\left(1-\theta\left(\mathrm{mod}(x+w,2)\right)\right)e^{-\Phi(e,x,w)}P_{tilt}(w,w')\\ +\,\theta(\mathrm{mod}(x+w,2))e^{-(1-O(e,x,w))\Phi(e,x,w)}P_{occl}(O(e,x,w),w,w')\end{array}\right] \quad (3.6.2)$$

$$\theta(y)=\begin{cases}1 & \text{if } y>0\\ 0 & \text{otherwise}\end{cases}$$

where $$\mathrm{mod}(x+w,2)=\begin{cases}1 & \text{if x+w odd}\\ 0 & \text{if x+w even}\end{cases}$$

$$w'=w-1,w,w+1$$

$$O(e,x,w)=\begin{cases}0 & \text{if } x+w \text{ even}\\ 0,1 & \text{otherwise}\end{cases}$$

3.6.1 Tilted Surfaces

A simple model for tilted surfaces is to introduce a constant bias for the tilted surface to differentiate from the flat surface solution. We propose:

$$P_{tilt}(w,w') = \frac{1}{C} e^{-TiltCost|w-w'|} \quad \overset{w'=w-1,w,w+1}{\rightarrow} \quad C = 1 + 2e^{-TiltCost} \tag{3.6.3}$$

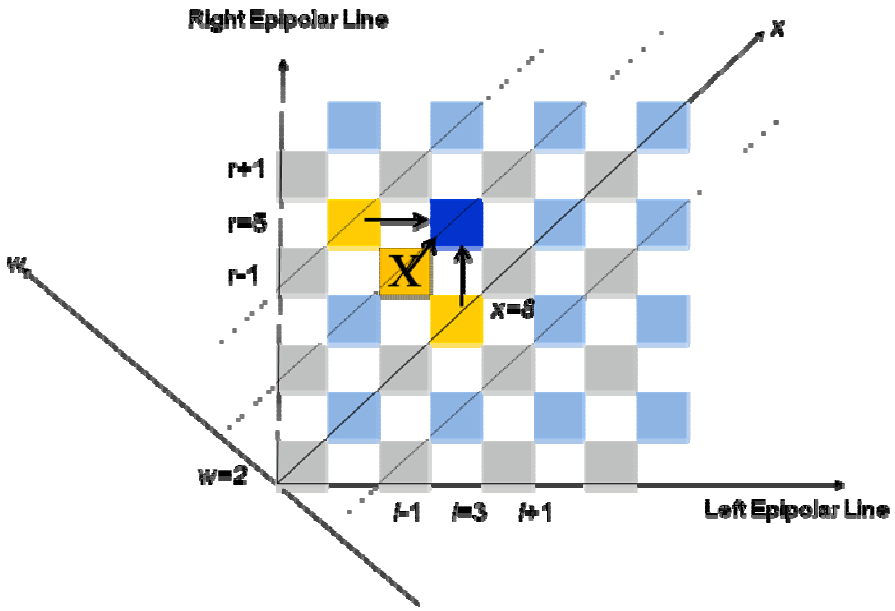


Figure 3.6.1: Tilted Surface Model. We assume the pixel pair $(l=3, r=5)$ or $(x=8, w=2)$ is a match (blue). Since $(x+w)$ is even, then the transition from $(x-1, w')$ (yellow) where $w' = w-1, w, w+1$ to (x, w) where $(x=8, w=2)$ (blue) is modeled as a flat surface for $w' = w$ (denoted as X, a sub-pixel match) and as a tilted surface for $w' = w-1, w+1$.

This prior model includes a normalized probability term $e^{-\Phi(e,x,w)}P_{tilt}(w,w')$ as

$$P_{tilt}(e,x,w,w') = \frac{e^{-\Phi(e,x,w)}}{Z_{tilt}} \begin{cases} e^{-(TiltCost)} & w' = w-1 \\ 1 & w' = w \\ e^{-(TiltCost)} & w' = w+1 \end{cases} \qquad (3.6.4)$$

3.6.2 Occluded Surfaces

We model occlusions as the path of nodes transitioning from a sub-pixel match to a sub-pixel match. We guarantee this representation by constraining that the field $O(e,x,w)$ can only take a value 1 if (e,x,w) represents a node of a sub-pixel to sub-pixel match, meaning that $(x+w)$ is odd (Equation 3.6.3). Moreover, when an occlusion transition occurs ($w \to w' = w-1, w+1$) we do not want to measure it by the data-match cost, $\Phi(e,x,w+D)$, since no match takes place, meaning $O(e,x,w)=1$. We then model $P_{occl}(O(e,x,w),w,w')$ as follows:

$$P_{occl}(O(e,x,w),w,w') = \frac{1}{Z_{occl}} \begin{cases} e^{-O(e,x,w)OcclusionCost\left(1-\eta\sqrt{\left|\frac{D^L(e,x,w)}{D^L_{\max}}\right|}\right)} & w' = w-1 \\ e^{-(1-O(e,x,w))} & w' = w \\ e^{-O(e,x,w)OcclusionCost\left(1-\eta\sqrt{\left|\frac{D^R(e,x,w)}{D^R_{\max}}\right|}\right)} & w' = w+1 \end{cases} \qquad (3.6.5)$$

Note that a fixed penalty is charged at the transitions, when $O(e,x,w)=1$. The penalty is proportional to an *OcclusionCost* parameter and can be diminished if an intensity edge exists. An intensity edge in the left eye, is an indication that a depth discontinuity is more likely to occur, seen from the left eye. The same holds true for the right eye. Thus the parameter η controls how much the likelihood of a disparity discontinuity varies due to the intensity edges. Finally, the normalized occlusion probability $e^{-(1-O(e,x,w))\Phi(e,x,w,s)}P_{occl}(O(e,x,w),w,w')$ is described by:

$$P_{occlusion}(e,x,w,w') = \frac{1}{Z_{occl}} \begin{cases} e^{-OcclusionCost\left(1-\eta\sqrt{\left|\frac{D^L(e,x,w)}{D^L_{\max}}\right|}\right)} & w' = w-1 \\ e^{-\Phi(e,x,w,s)} & w' = w \\ e^{-OcclusionCost\left(1-\eta\sqrt{\left|\frac{D^R(e,x,w)}{D^R_{\max}}\right|}\right)} & w' = w+1 \end{cases} \qquad (3.6.6)$$

where the decision of $O(e,x,w)=1$ always occurs for $(x+w)$ *odd* and $|w-w'|=1$, meaning that an occlusion occurs for these odd nodes when a change of disparity occurs. Thus, (Equation 3.6.6) does not need to explicitly consider the occlusion field, since its optimal values are already known (in all cases).

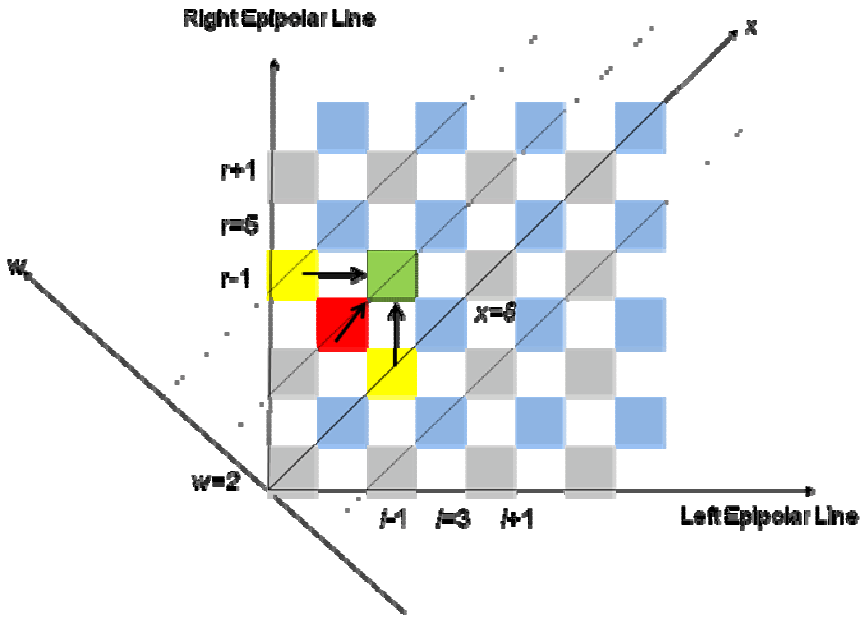


Figure 3.6.2: Occluded Surface Model. This figure shows the occlusion path from $(x-1, w')$ to (x, w) indicated by the three arrows. We assume that the pixel pair at the green square is a match for (x, w). The transition through constant disparity (flat surface) is shown coming from the red square $(w' = w)$, while the transition through an occlusion is shown via the yellow squares $(w' = w-1, w+1)$.

3.6.3 Posterior Model

Our posterior model is then

$$P(\{w(x,e)\} \mid I^L, I^R) =$$
$$\frac{1}{Z}\prod_{e=0}^{N-1}\prod_{x=1}^{2N-1}\left[\left(1-\theta\left(\text{mod}(x+w,2)\right)\right)P_{tilt}(e,x,w,w') + \theta(\text{mod}(x+w,2))P_{occlusion}(e,x,w,w')\right]$$

$$\theta(y) = \begin{cases} 1 & \text{if } y > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $$\text{mod}(x+w,2) = \begin{cases} 1 & \text{if x+w odd} \\ 0 & \text{if x+w even} \end{cases}$$

$$w' = w-1, w, w+1$$

The optimal values of the occlusion field $O(e,x,w)$ have already been estimated. More precisely, $O(e,x,w) = 0$ everywhere except at $(x+w)$ *odd* and $|w-w'| = 1$ where $O(e,x,w) = 1$.

3.7 Epipolar Line Interactions

There is sufficient evidence supporting that the disparity solution produced by human vision reflects interdependence among disparity values at different epipolar lines. The illusory rectangle stereo pair, shown in Figure 3.7.1 below, provides such evidence. In the middle of the rectangle, a white line in the left image (without any distinct feature) matches a white line in the right image producing a disparity $w(e, x)$ that varies along x, which reflects the interdependence of the other epipolar line solutions. The white line solution "gets carried" by the other neighboring epipolar line solutions.

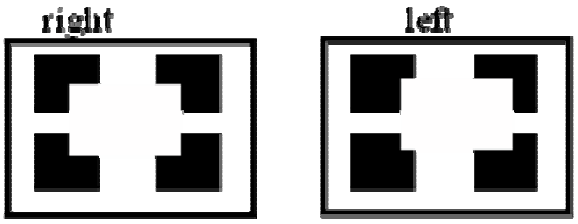


Figure 3.7.1: Stereo Pair of an Illusory Rectangle. Considering the stereo pair of a white square in front of 4 black squares, there is a vertical interaction of epipolar lines. The larger the intensity edges along the vertical axis are, the higher the probability of have a disparity change across epipolar lines.

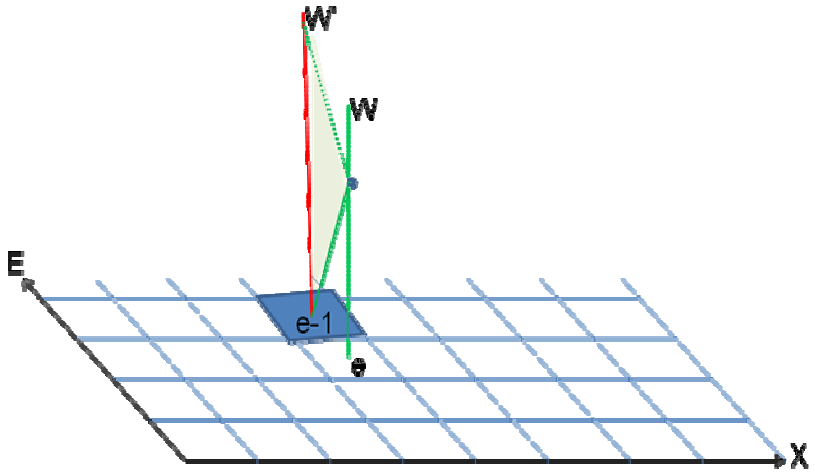


Figure 3.7.2: Vertical Epipolar Line Interaction. Considering the vertical interaction between epipolar lines in the cyclopean coordinate system, a coordinate (e,x) and w can have a relationship with all w' of the previous coordinate $(e-1,x)$. In this case, $w'=-D\ldots+D$, instead of equation $(|w(e,x)-w(e,x-1)|\leq 1)$ in the case of the smoothness constraint for the horizontal interactions.

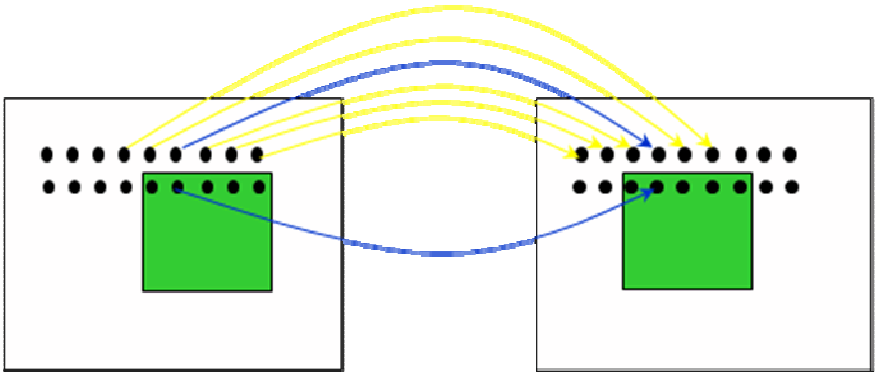


Figure 3.7.3: Modeling Disparity across Epipolar Lines. When modeling the vertical interactions across epipolar lines, in order to model disparity changes between (e, x) with the previous epipolar line $(e-1, x)$ we must match the $l+i$ values in one image with the $r+i$ values in the other image and vice versa. This is necessary since fixing x and varying w requires that the l and r values in both images need to be compensated.

In order to model the epipolar interaction we consider that the larger the intensity edges across epipolar lines, the less of a cost there is to have a disparity change across the epipolar lines. We also allow for any disparity change across epipolar and do not enforce a vertical disparity gradient in our model, as is shown in Figure 3.7.2. In order to model vertical disparity changes across epipolar lines, it is worth noting that window matching in the left and right stereo pair is reversed between the two images. Figure 3.7.3 explains this property of reversed window matching when fixing x and varying w due to equations (3.2.1, 3.2.2). With this being said, we propose the following model.

$$P_{epipolar}\left(\{w(x,e)\}\right)=\prod_{e=1}^{N-1}\prod_{x=0}^{2N-1}P_{epip}\left(w''(x,e-1),w(x,e)\right)$$

$$P_{epip}\left(w''(x,e-1),w(x,e)\right)=\frac{1}{C}e^{-\mu|w-w''|(1-\eta VI_{e,x,w,w'})}$$

$$VI_{e,x,w,w'}=\max\left[\left(\frac{\left|\hat{I}^L(l(x,w),e,\frac{3\pi}{2},5)-\hat{I}^L(l(x,w''),e-1,\frac{\pi}{2},5)\right|+1}{\hat{I}^L(l(x,w),e,\frac{3\pi}{2},5)+\hat{I}^L(l(x,w''),e-1,\frac{\pi}{2},5)+1}\right),\left(\frac{\left|\hat{I}^R(r(x,w),e,\frac{3\pi}{2},5)-\hat{I}^R(r(x,w''),e-1,\frac{\pi}{2},5)\right|+1}{\hat{I}^R(r(x,w),e,\frac{3\pi}{2},5)+\hat{I}^R(r(x,w''),e-1,\frac{\pi}{2},5)+1}\right)\right]$$

Our final model for the probabilities given is:

$$P(\{w(x,e)\}\,|\,I^L,I^R)=$$
$$\frac{1}{Z}\prod_{e=1}^{N-1}\prod_{x=1}^{2N-1}\left[\begin{array}{l}\left(1-\theta\left(\mathrm{mod}(x+w,2)\right)\right)P_{tilt}(e,x,w,w')+\\ \theta(\mathrm{mod}(x+w,2))P_{occlusion}(e,x,w,w')\end{array}\right]P_{epip}(w,w'')$$

$$\theta(y)=\begin{cases}1 & \text{if } y>0\\ 0 & \text{otherwise}\end{cases}$$

where $$\mathrm{mod}(x+w,2)=\begin{cases}1 & \text{if x+w odd}\\ 0 & \text{if x+w even}\end{cases}$$

$$w'=w-1,w,w+1$$

This model has the following parameters that need to be estimated: T, *OcclusionCost*, *TiltCost*, η, and μ. We will introduce the section describing the experiments by studying the role and value for these parameters. The robustness of

this model is assessed by the range of parameter values that yield good solutions. By studying difficult imagery, one can also study the power of the model.

3.8 Belief Propagation Implementation with BP-TwoGraphs

Our final model of the probabilities is given as:

$$P(\{w(x,e)\} \mid I^L, I^R) =$$
$$\frac{1}{Z}\prod_{e=1}^{N-1}\prod_{x=1}^{2N-1}\begin{bmatrix}\left(1-\theta\left(\text{mod}(x+w,2)\right)\right)P_{tilt}(e,x,w,w')+\\ \theta(\text{mod}(x+w,2))P_{occlusion}(e,x,w,w')\end{bmatrix}P_{epip}(w,w'')$$

This can then be written in a form to apply a belief propagation scheme as:

$$P(\{w(x,e)\} \mid I^L, I^R) = \frac{1}{Z}\prod_{e=1}^{N-1}\prod_{x=1}^{2N-1}\psi\left(w_{e,x}, w'_{e,x-1}, w''_{e-1,x}\right)$$

where

$$\psi\left(w_{e,x}, w'_{e,x-1}, w''_{e-1,x}\right) =$$
$$\begin{bmatrix}\left(1-\theta\left(\text{mod}(x+w,2)\right)\right)P_{tilt}(e,x,w,w')+\\ \theta(\text{mod}(x+w,2))P_{occlusion}(e,x,w,w')\end{bmatrix}P_{epip}(w,w'')$$

Therefore, the belief propagation scheme, equation (2.5.1), is then set for the ψ above and $\phi_i(x_i, y_i) = 1$. We can also divide the ψ term into a vertical interaction term and horizontal interaction term as follows:

$$\psi\left(w_{e,x}, w'_{e,x-1}, w''_{e-1,x}\right) = \psi_H\left(w_{e,x}, w'_{e,x-1}\right)\psi_V\left(w_{e,x}, w''_{e-1,x}\right)$$

where

$$\psi_H\left(w_{e,x}, w'_{e,x-1}\right) = \begin{bmatrix} \left(1-\theta\left(\mathrm{mod}(x+w,2)\right)\right)P_{tilt}(e,x,w,w') + \\ \theta(\mathrm{mod}(x+w,2))P_{occlusion}(e,x,w,w') \end{bmatrix}$$

$$\psi_V\left(w_{e,x}, w''_{e-1,x}\right) = P_{epip}(w,w'')$$

This decomposition is useful for the belief propagation scheme described in Chapter 2.5.

4 Experiments and Results

We will present the different stages and evolution of the STUMP algorithm. First we will explain our preliminary setup for the experiments, the parameters which we used for our model, a walkthrough of our implementation, and an explanation of our results. Following the setup of our experiments, will be series of different experiments categorized into three rounds. We will show our solution of STUMP for various stereo images, illustrate parameter configurations, and finally display a comparison of STUMP with an implementation of a dynamic programming algorithm.

The complexity of the STUMP algorithm can be estimated in terms of the number of pixels of the stereo pair of images (width by height), the range of disparity, and the number of scales used in extracting features. According to the complexity discussed by Lui [17], an estimation of the complexity can be stated at $\alpha NS^2 + \beta NS$, where (α, β) are two constant coefficients, N is the total number of pixels of the cyclopean coordinate system, (e, x), and S is the range of disparity from $-D \; to \; +D$. The complexity the feature extraction is $\gamma N'L$, where N' is the number of pixels of the images in the stereo pair and L is the number of scales used, and is a γ constant. The complexity for the ϕ and ψ functions of the STUMP algorithm are δNSL and

εNS^2, where (δ, ε) are two constant coefficients. Combining all of the complexities as one, we can introduce a quadratic polynomial complexity as: $\alpha' NS^2 + \beta' NS + \gamma' NSL + \delta' NL$, where $(\alpha', \beta', \gamma', \delta')$ are constant coefficients. The number of scales used is not large and usually varies between 1 and 10, and therefore we can simply say that the STUMP algorithm is $O(NS^2)$.

4.1 Settings

In order to present a full detailed explanation of the experiments conducted on the STUMP algorithm, we must first introduce the preliminary stages used to construct the algorithm. At a first glance, our algorithm takes a pair of stereo images as input. Specifying two images of the same dimensions and resolutions, already with epipolar lines configured along the vertical axis of the images, we begin to perform data extraction.

The basic implementation of extracting data, which can also be understood as detecting features among the stereo pair of images, is via differences in intensities at pixel levels as well as edge detection at a sub-pixel level. As mentioned in the previous chapter, the Φ function captures the data-matching aspect of the algorithm. Initially, each pixel of each image undergoes intensity measurements by computing the difference in the intensities for neighboring pixels horizontally and vertically.

Also as part of the initialization process, a given scale is defined in which the number of pixels measured is averaged. For the first round of experiments we set the scale parameter to be 3 pixels, meaning that for each pixel (i, j), the intensity computed was an average of $\sum_{s=0}^{scale} \text{intensity}(i+s, j+s)$ in the directions (orientations) for a given $\theta = \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$. The Φ function, as defined previously, performs data-matching based upon average intensity differences along with edge detection. Edge detection is determined by the differences in derivatives for the intensities measured for a given pixel. Due to the mapping of pixels (i, j) into the cyclopean system of (e, x), half of the values in the cyclopean coordinate system occur at the pixel level, while the other half occur at the sub-pixel level. At a sub-pixel level the intensity measurements for a given pixel (i, j) is averaged among the intensity computation for pixels (i, j) and the previous (i, j) with respect to a given horizontal or vertical axis. The derivative at a given location (i, j), in which (i, j) is at the pixel level, is computed as the measured intensity difference between the previous (i, j) and the next (i, j). However, at the sub-pixel level the derivative is determined by the difference between (i, j) and the previous (i, j).

Now that data can be measured and extracted from our stereo image input, we will address how to process our data via adjusting well-defined parameters. During the following test round, our parameters of interest include $\{D, \eta, tilt, occlusion, \mu\}$.

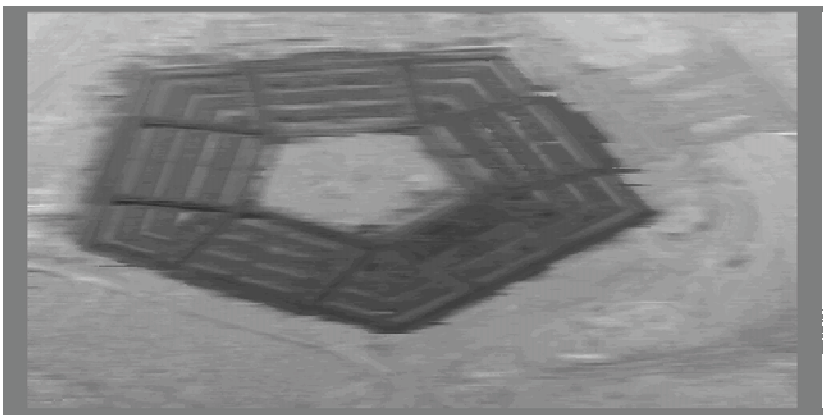
The parameter D is defined as the disparity range for a given experiment. In all experiments the disparity ranges from $-D$ to $+D$. The disparity range determines how near or far objects are in our disparity map result-image. Negative disparity values map to darker pixels where as positive disparity values map to lighter pixels within the disparity map result-image. For all experiments, there is a slight bias along the border of the image towards disparity zero. The parameter η is defined as the "edge-ness" for our ψ function for the areas in which edges are considered. Within our ψ function, edges are considered for odd values of $(x+w)$, since in our cyclopean coordinate system these "odd" values correspond to sub-pixels which can be understood as edges between pixels. The parameter η can range from 0 to 1, although for most experiments we fix η at 0.6. The parameters *tilt* and *occlusion* are constant coefficients given to the ψ function for areas in which there is thought to be either a tilted surface or an occluded surface. These values vary from experiment to experiment. Finally the parameter μ defines how much of a vertical interaction should there be between epipolar lines. The parameter μ may vary in experiments with respect to changes of *tilt* and *occlusion*. Now that our initial setup is presented and the parameters are explained, we will proceed to display the results of the following experiments.

The cyclopean coordinate assigns disparity to subpixel-subpixel matching as well as pixel-pixel matching and therefore has twice as many points. When displaying

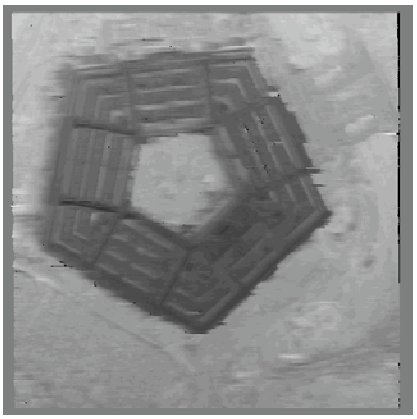
the disparity map STUMP results, it appears that the image is stretched, due to the extra sub-pixel values. It is worth noting that the epipolar lines for the cyclopean coordinate system are the same, thus stretching does not appear along epipolar lines.

Our experiments output disparity solutions in the cyclopean system. However, after the resulting disparity map is computed, the disparity map can undergo a transformation back into the image coordinate system, if one wishes to see how the result would appear in either the left or right eye. This is shown during the first round of experiments as well as in Figure 4.1.1 below.

a)



b)



c)

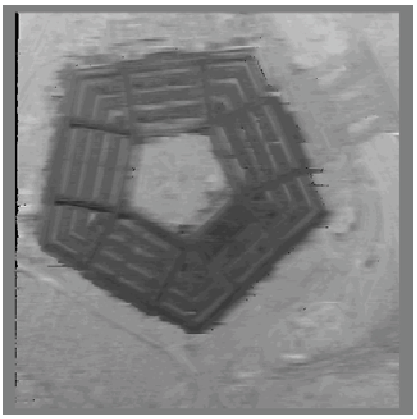


Figure 4.1.1: Cyclopean Disparity Map as Seen From the Left and Right Eyes. Image (a) shows the disparity map for the STUMP result for the pentagon stereo pair as shown in Experiment 1.3. Image (b) is the same disparity map solution of image (a) but transformed back into the left coordinate system to show a disparity solution as seen by the left eye. Likewise, the same is done for the right eye in image (c). It is also worth noting that the black pixels (difficult to see at the resolution of this paper) of images (b) and (c) are points in which there is no solution in the left or right coordinate system (l, r) for a solution of the cyclopean coordinate system (x, w).

In the event that one wishes to see the cyclopean disparity map as seen from either the left or right eye, one must realize that not all the (x, w) values of the cyclopean solution will translate back to values in the (l, r) coordinate system. This being said, one can notice the black pixels (although difficult to see at the resolution of this paper) in the images (b) and (c) of Figure 4.1.1. In order to display a 3D result of either the left or right source image overlaid over the disparity map, the pixels in which there is no solution in the left or right disparity map are averaged. This procedure is only intended to show a 3D representation (in the first round of experiments) of how one would see an image (either left or right) given a disparity map. Our intended purpose of the STUMP algorithm is to provide a disparity map in the cyclopean eye, not the left and right eyes independently.

4.2 Results

Our first round of experiments will show the disparity map for a stereo pair of images using the STUMP algorithm. Following this result for each stereo pair will be a second round of experiments to show how the parameters for each test are configured. This second round will both display the effect of each parameter as well show the robustness of our algorithm as we vary each parameter. Finally, there will be a third round of experiments in which the result of STUMP algorithm for each stereo pair of images is compared to the results of a dynamic programming implementation.

This round demonstrates the difference of dynamic programming compared to STUMP, both with and without vertical interactions.

Experiment Round 1: STUMP Disparity Map Results for Various Pairs of Stereo Images

Experiment 1.1: Random-Dot Stereogram in which $D=6$, $\eta=0$, $tilt=50, occlusion=100$, $\mu=50$, and a weighted scale $s=\{1,2,3,4,5,6,7\}$. This stereogram was constructed in the same manner as the stereogram presented previously in Chapter 2, with a displaced square area of pixels in the center. Figure (a) shows the original input for the 64x64 pixel left and right images of the random dot stereogram. Figure (b) is the result of running the STUMP algorithm. Since there is a centered square region of pixels displaced by 4 pixels, fusing the two images will yield a 3D square of random dots emerging from the random dot background. Figure (b) shows the background at a constant disparity further back (light gray) and a 3D square of constant disparity in the foreground (dark gray). Figure (c) shows the translation of the disparity map from the cyclopean coordinate system back into left and right images as scene by each eye. Finally, figure (d) shows the left image of figure (a) combined with the disparity map for the left image of figure (d) to show a 3D version

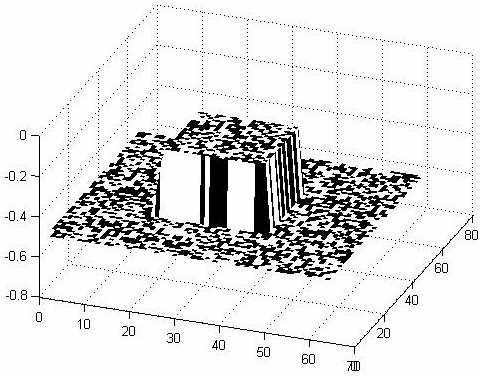
of the fused random dot image as seen from the left eye. The same can also be done for the right image, although it is not shown.

a)

b)

c)

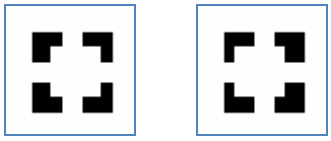
d)



Experiment 1.2: Illusory Square in which $D = 6$, $\eta = 0.5$, $tilt = 30$, $occlusion = 40$, $\mu = 45$, and a weighted scale $s = \{1,2,3\}$. Once again figure (a) shows the left and

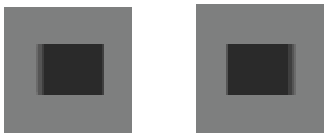
right stereo pair for an image of 4 black squares in the background and a single white square in the foreground displaced 4 pixels. Fusing both images yields a 3D white square in the foreground and the 4 black squares in the background. Figure (b) shows the results of the disparity map from the STUMP algorithm. Our algorithm shows the correct solution for the disparity of the illusory square due to epipolar line interaction. Although the epipolar lines in the center do not contain any data supporting the 3D square, the vertical interaction of STUMP connect the top of the white square with the bottom of the white square to give the illusion of a 3D square, although neither image contains a white square. It is also worth noting that the left and right edges of the square display a disparity gradient which is modeled by the smooth surface constraint, $|w(e,x)-w(e,x-1)|\leq 1$ (Chapter 3.5). While this disparity gradient is modeled, it is not strictly enforced due to the belief propagation. Figure (c) shows the translation of the disparity map from the cyclopean coordinate system back into left and right images as seen by each eye. Finally figure (d) shows the left image of figure (a) combined with the disparity map for the left image of figure (d) to show a 3D version of the fused illusory square.

a)

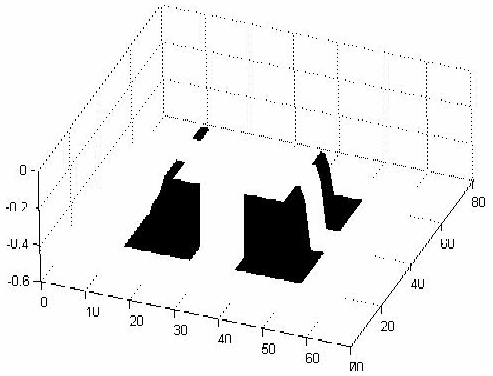


b)

c)



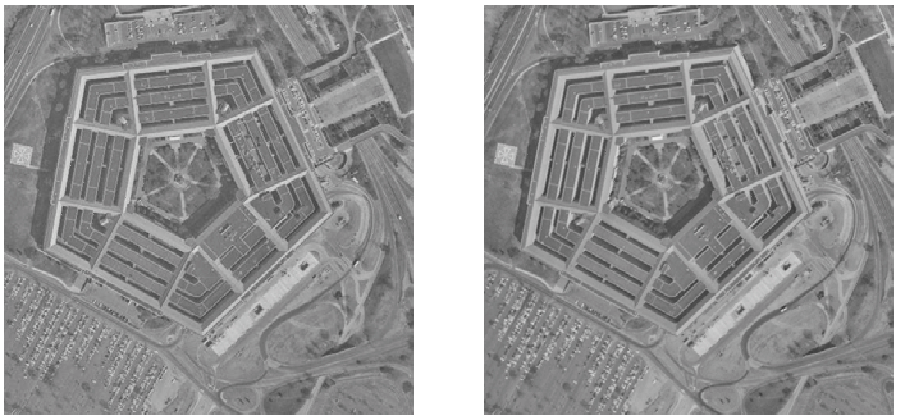
d)



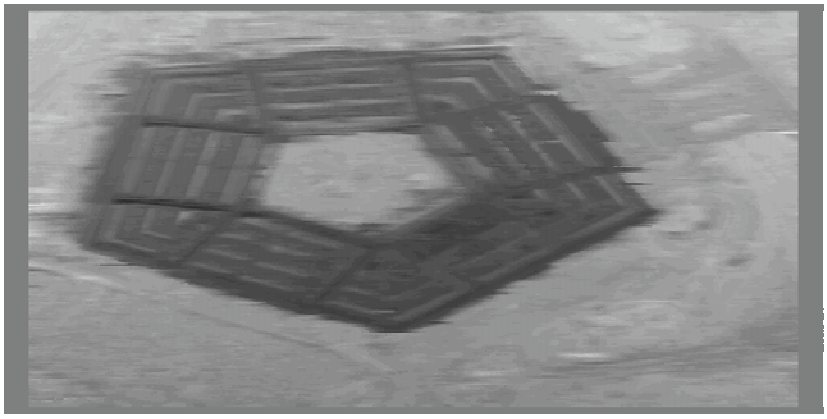
Experiment 1.3: Pentagon in which $D = 15$, $\eta = 0.5$, $tilt = 10$, $occlusion = 40$, $\mu = 1$ and a weighted scale $s = \{1,2,3,4,5,6,7\}$. This experiment shows how the STUMP algorithm performs on real world images. The pentagon stereo pair is a pair of grayscale 512x512 images. Figure (a) shows the left and right stereo pair of input images. Figure (b) shows the result of the disparity map for STUMP. One can see clearly that the pentagon is at a different disparity from the ground. Figure (c) shows the translation of the disparity map from the cyclopean coordinate system back into left and right images as seen by each eye. Finally figure (d) shows the left image of

figure (a) combined with the disparity map for the left image of figure (d) to show a 3D version of the fused pentagon.

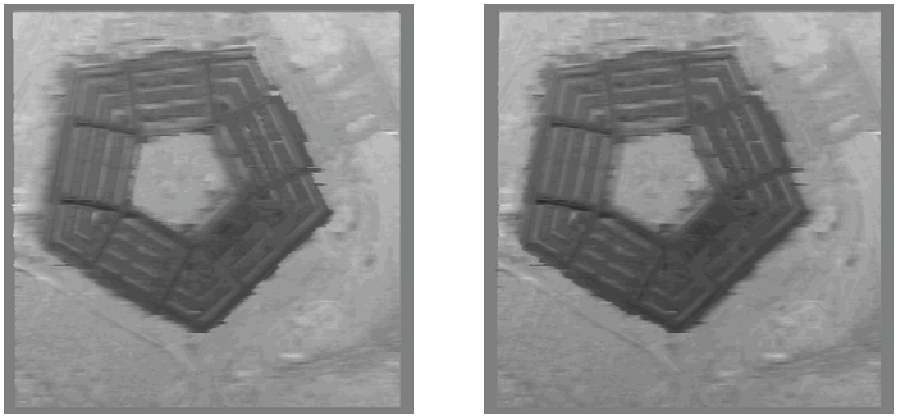
a)

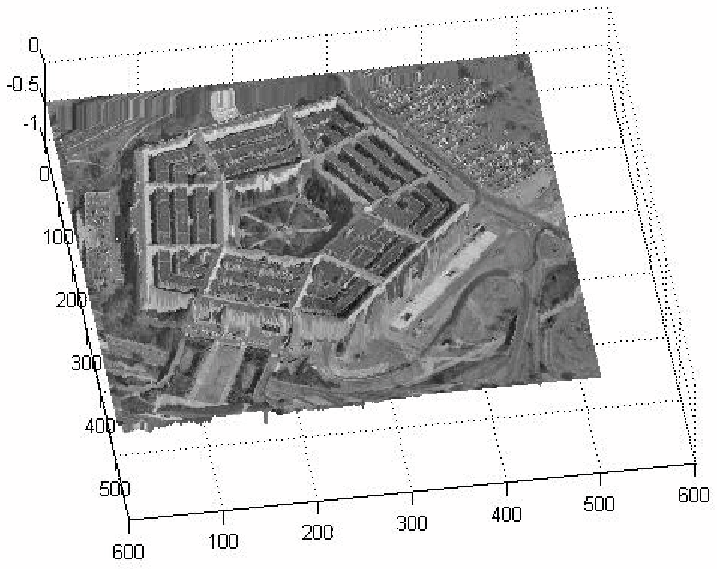


b)



c)

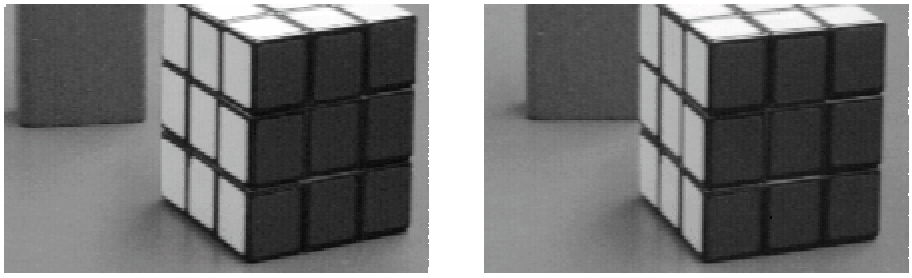




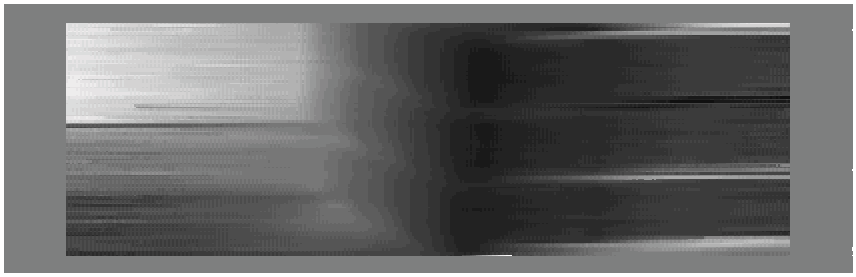
d)

Experiment 1.4: Cube in which $D = 15$, $\eta = 0.5$, $tilt = 60$, $occlusion = 80$, $\mu = 2$, and a weighted scale $s = \{1,2,3,4,5,6,7\}$. This experiment also shows how the STUMP algorithm performs on real world images. The cube stereo pair is a pair of grayscale 212x134 images. Figure (a) shows the left and right stereo pair of input images. Figure (b) shows the result of the disparity map for STUMP. Figure (c) shows the translation of the disparity map from the cyclopean coordinate system back into left and right images as seen by each eye. Finally figure (d) shows the left image of figure (a) combined with the disparity map for the left image of figure (d) to show a 3D version of the fused cube.

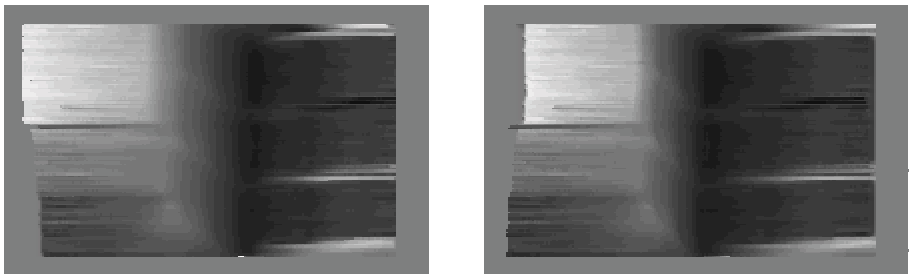
a)



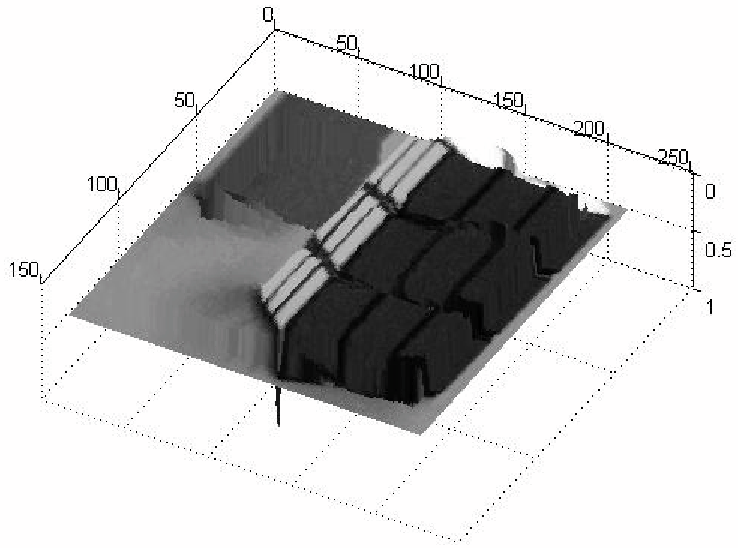
b)



c)



d)



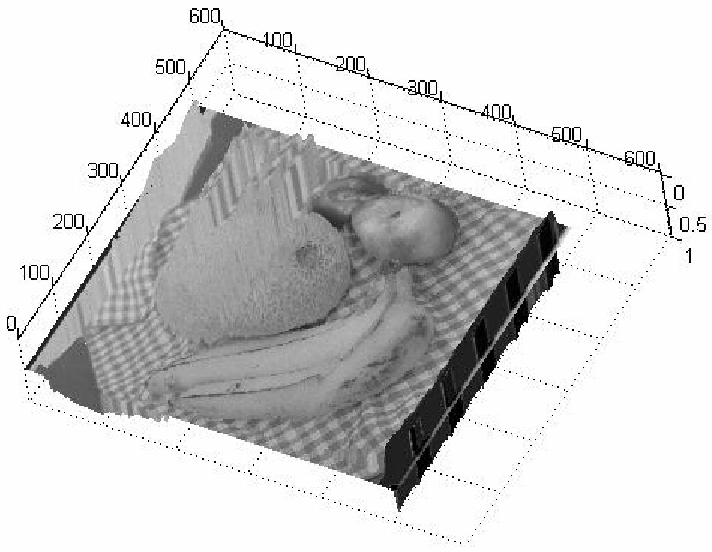
Experiment 1.5: Fruit in which $D = 25$, $\eta = 0.5$, $tilt = 60$, $occlusion = 40$, $\mu = 2$, and a weighted scale $s = \{1,3,5,7\}$. This experiment also shows how the STUMP algorithm performs on real world images. The fruit stereo pair is a pair of grayscale 512x512 images. Figure (a) shows the left and right stereo pair of input images. Figure (b) shows the result of the disparity map for STUMP. Due to the fact the stereo pair of images contains bad data (such as the black speck in the right image), the disparity map solutions are noisy and capture the mismatches of bad data. Figure (c) shows the translation of the disparity map from the cyclopean coordinate system back into left and right images as seen by each eye. Finally figure (d) shows the left image of figure (a) combined with the disparity map for the left image of figure (d) to show a 3D version of the fused fruit.

a)

b)

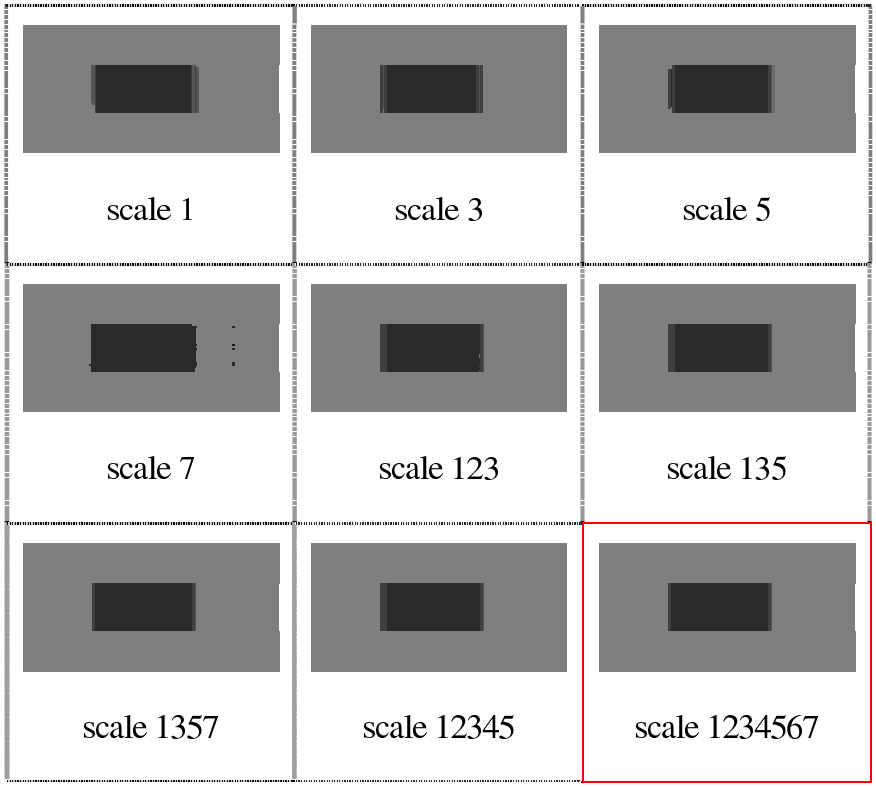
c)

d)

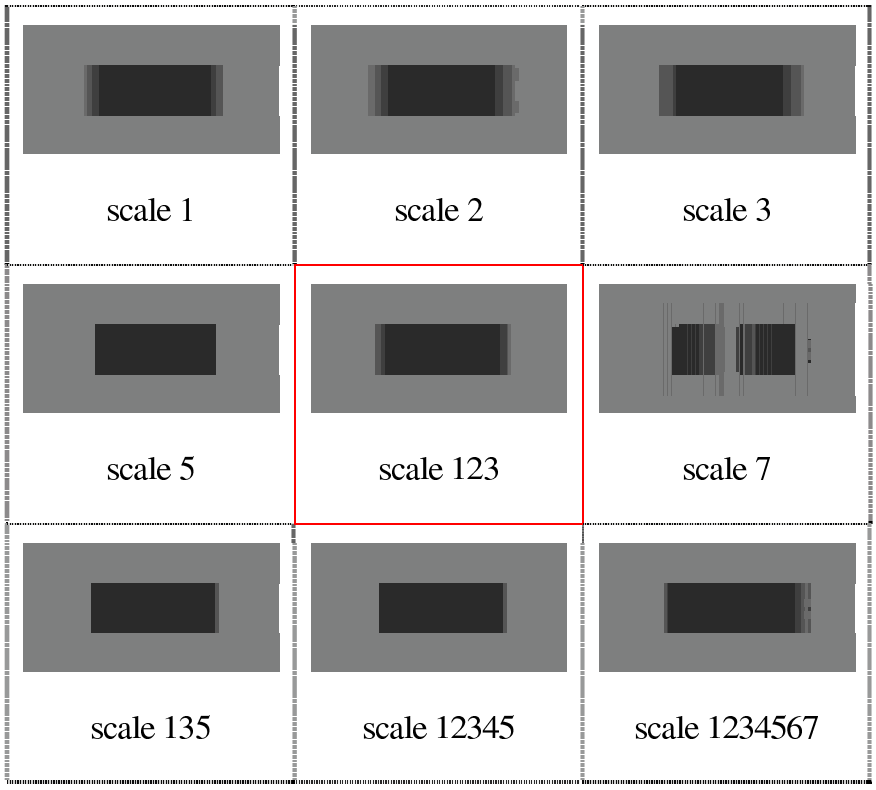


Experiment Round 2: STUMP Parameter Configurations

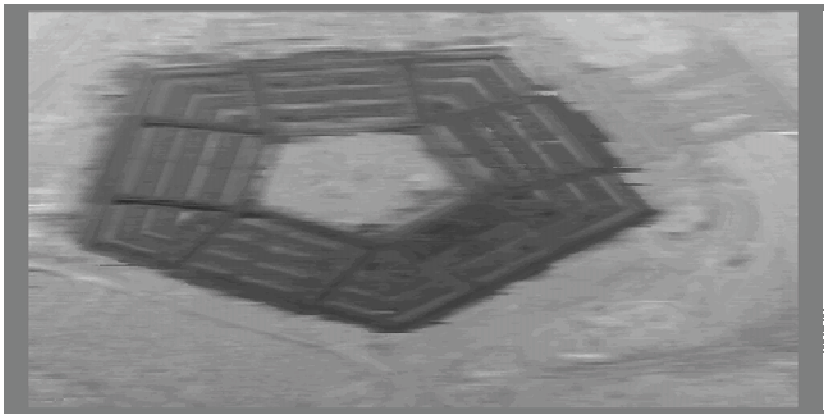
Experiment 2.1.1: Configuring the *scale* parameter for the random dot stereo pair from Experiment 1.1. The bottom right image with $scale = 1,2,3,4,5,6,7$ (red) is the default scale parameter as used in Experiment 1.1. All other images of the following table show the results of the disparity map from the STUMP algorithm with different scales. Some images use a single scale while others use a combined weighted scale for Φ as explained previously in Equation (3.3.20). This illustrates our entropy model discussed in (Chapter 3.3.1), in which one can see the average of $scale = 1$, $scale = 3$, and $scale = 5$ combined in the result of $scale = 1,3,5$.

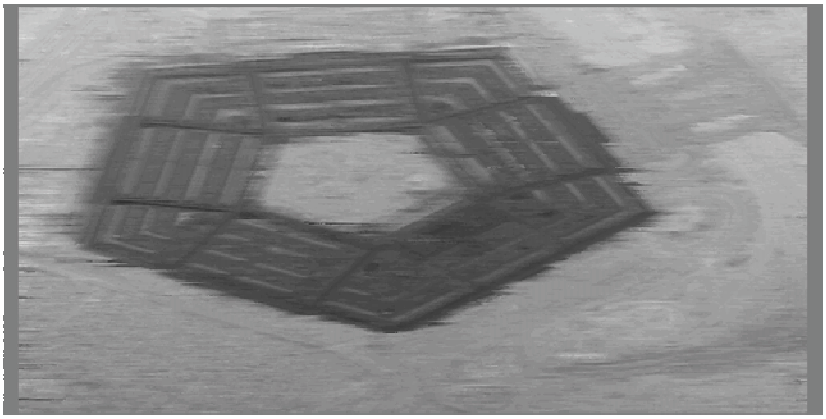


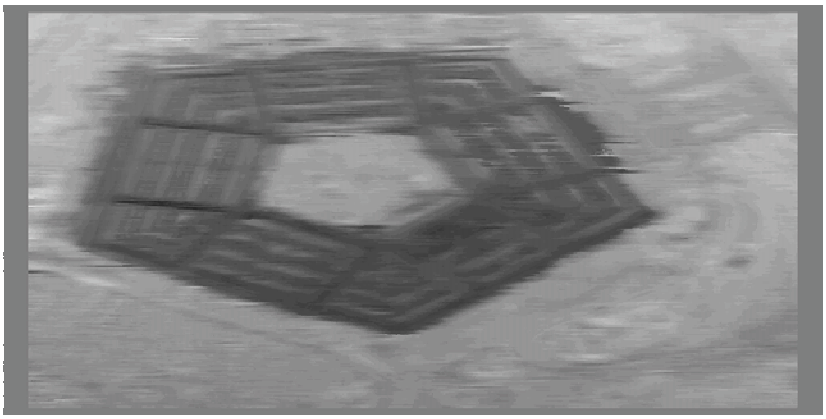
Experiment 2.1.2: Configuring the *scale* parameter for the illusory square stereo pair from Experiment 1.2. The middle image with $scale = 1,2,3$ (red) is the default scale parameter as used in Experiment 1.2. All other images of the following table show the results of the disparity map from the STUMP algorithm with different scales. Some images use a single scale while other use a combined weighted scale for Φ as explained previously in Equation (3.3.20). The image with $scale = 1,2,3$ was chosen as the default since it modeled the disparity gradient for epipolar lines at the boundaries of the occluded square well. Varying the *scale* parameter also illustrates our entropy model discussed in (Chapter 3.3.1), in which one can see the average of $scale = 1$, $scale = 2$, and $scale = 3$ combined in the result of $scale = 1,2,3$.



Experiment 2.1.3: Configuring the *scale* parameter for the pentagon stereo pair from Experiment 1.3. The top image (a) with $scale = 1,2,3,4,5,6,7$ is the default scale parameter as used in Experiment 1.4. Image (b) shows $scale = 1$ while image (c) shows $scale = 7$. The image with $scale = 1,2,3,4,5,6,7$ was chosen as the default since it averages the images (b) and (c) rather well, although it is difficult to see pixel differences at the resolutions printed on paper.

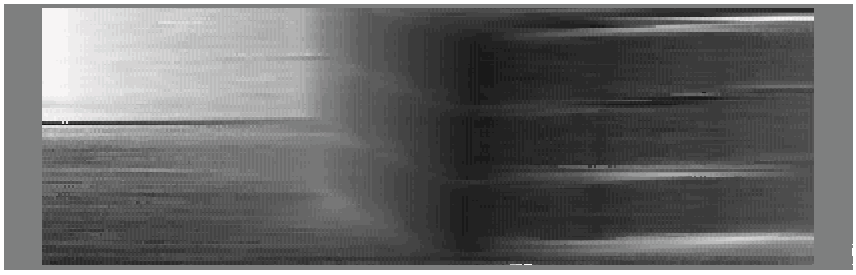
a) 

b) 

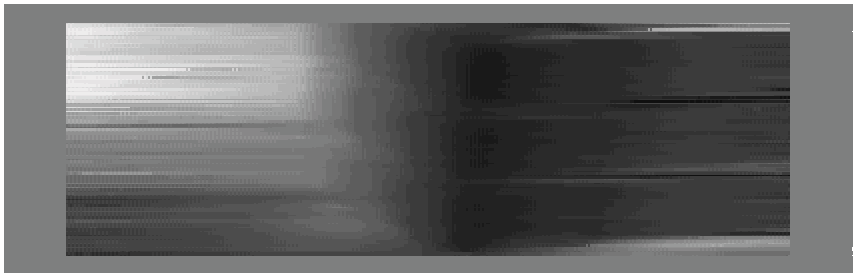
c) 

Experiment 2.1.4: Configuring the *scale* parameter for the cube stereo pair from Experiment 1.4. The top image (a) with $scale = 1$. Image (b) shows $scale = 7$ while image (c) shows a weighted $scale = 1,2,3,4,5,6,7$, which is the default scale parameter as used in Experiment 1.4. The image with weighted $scale = 1,2,3,4,5,6,7$ was chosen as the default since it averages the images (a) and (b) rather well using our entropy model. Finally, image (d) shows a weighted scale of $scale = 1,3,5,7,9,11,13,15$.

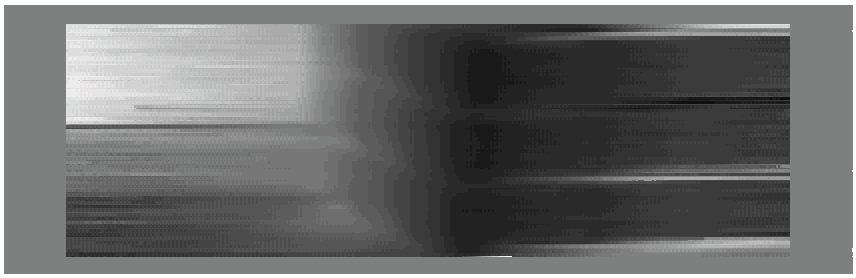
a)



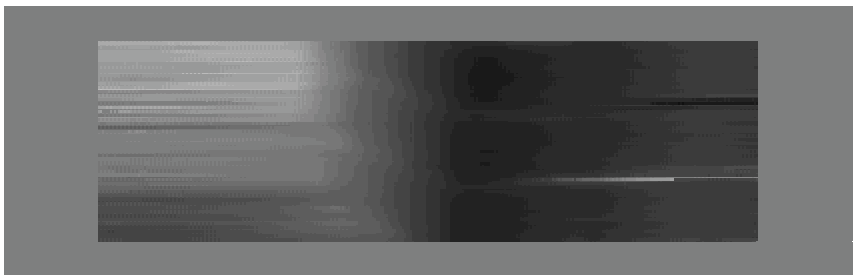
b)



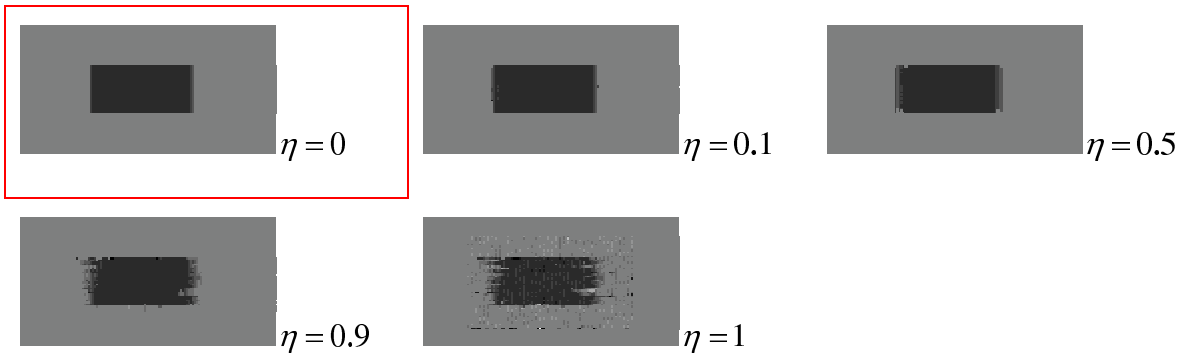
c)



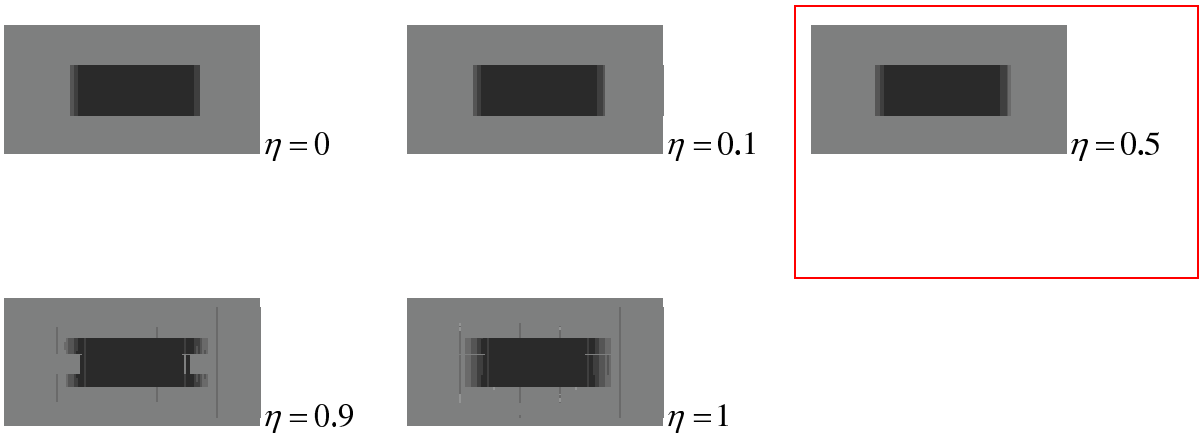
d)



Experiment 2.2.1: Configuring the η parameter for the random dot stereo pair from Experiment 1.1. The first image below $\eta = 0$ (red box), is the default image chosen since the random dot stereo pair does not take into account any edge since no intended edges exist. By varying η, one can see its role in the STUMP algorithm. High values of η will detect much unwanted noise rather than intended edges.



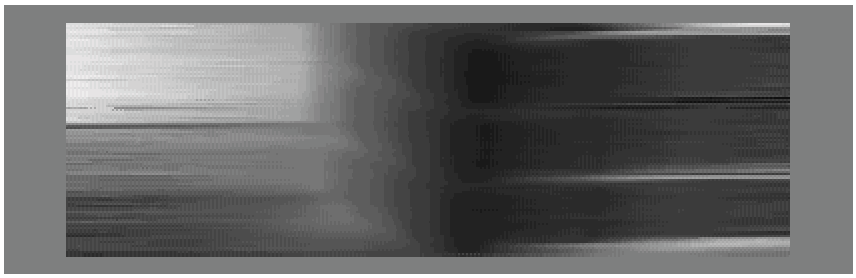
Experiment 2.2.2: Configuring the η parameter for the illusory square stereo pair from Experiment 1.2. The image below $\eta = 0.5$ (red box), is the default image chosen since we wish not to make a bias for or against edges, given that edges do exists in the stereo pair. By varying η, one can see its role in the STUMP algorithm and how STUMP is quite robust for a range of η.



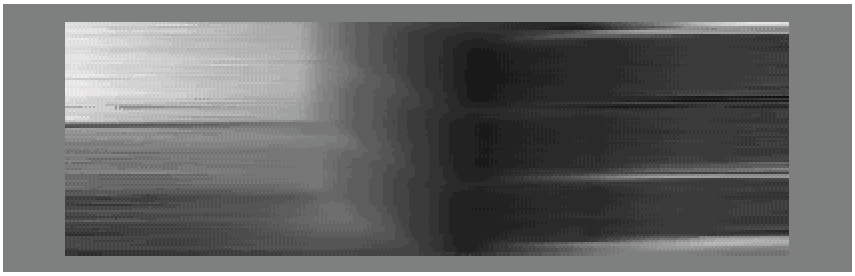
Experiment 2.2.3: Configuring the η parameter for the cube stereo pair from Experiment 1.4. The image (c) below $\eta = 0.5$, is the default image chosen since we wish not to make a bias for or against edges. By varying η, one can see its role in the STUMP algorithm and how STUMP is quite robust for a range of η. The effect of the

parameter η does not influence real world images as much as our test images since there is an abundant amount of data in the real world images.

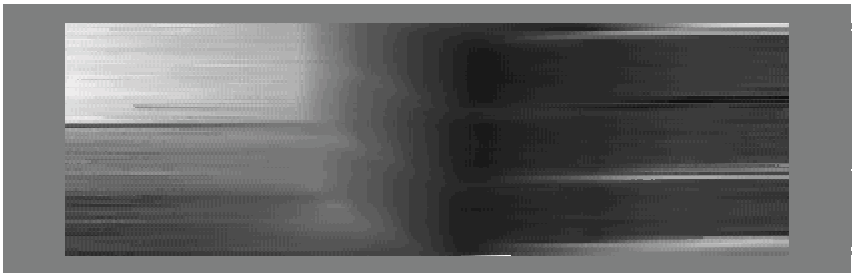
a) $\eta = 0$



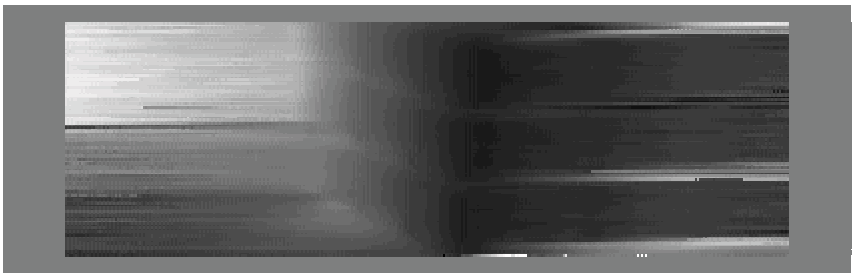
b) $\eta = 0.1$

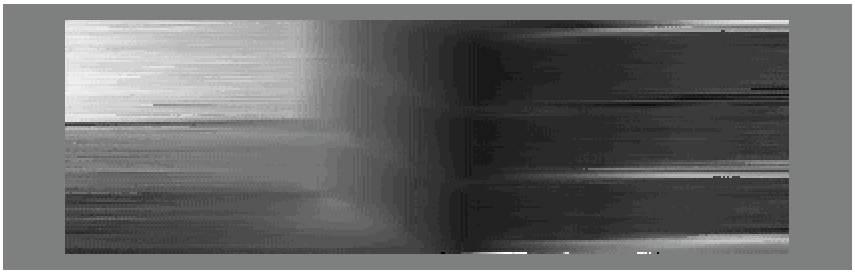


c) $\eta = 0.5$

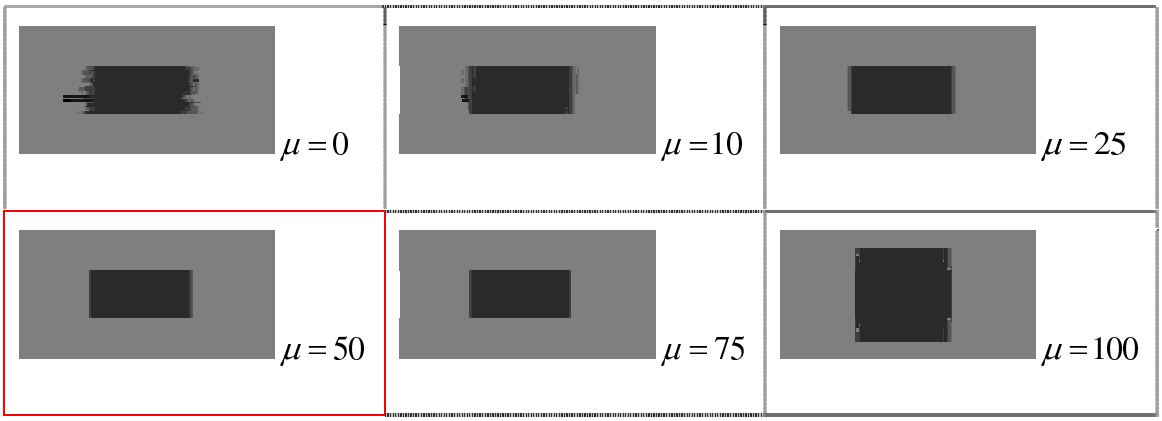


d) $\eta = 0.9$

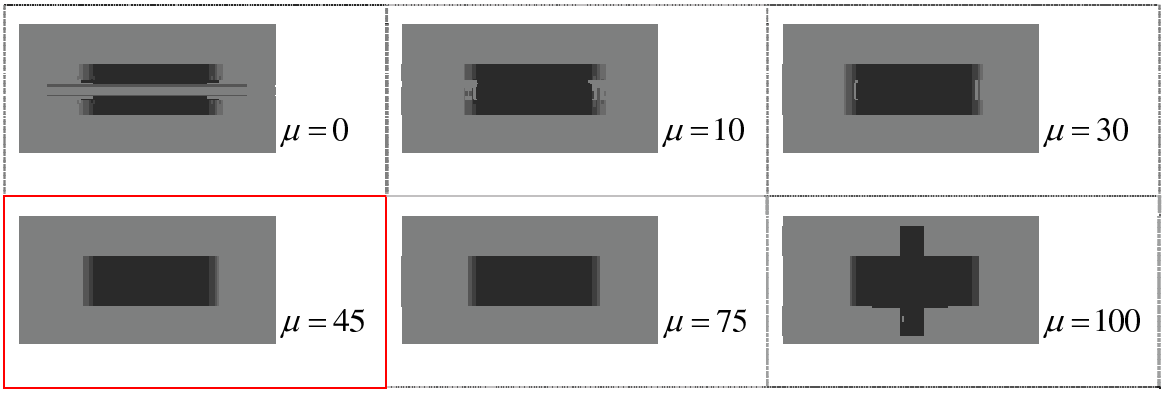


e) $\eta = 1.0$ 

Experiment 2.3.1: Configuring the μ parameter for the random dot stereo pair from Experiment 1.1. The image below, $\mu = 50$, was chosen as the default value for the μ parameter for the random dot stereo pair since it is the lowest value which strengthens the vertical interaction of the epipolar lines. One can also see that the μ parameter is quite robust, although upon a certain threshold, the vertical interaction becomes too overbearing (such as in the case of $\mu = 100$).

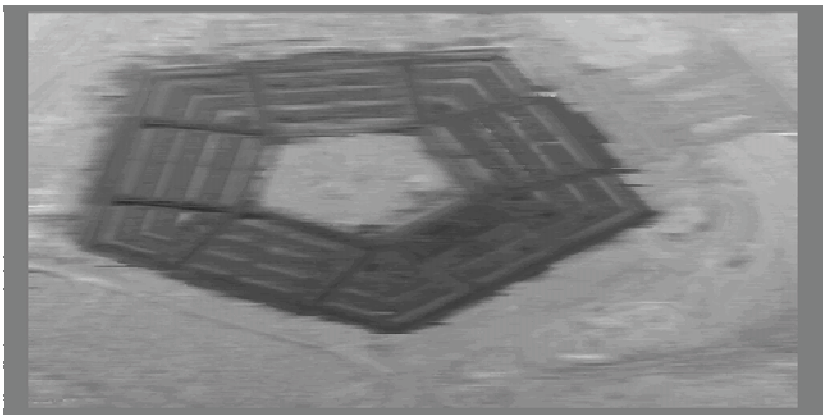


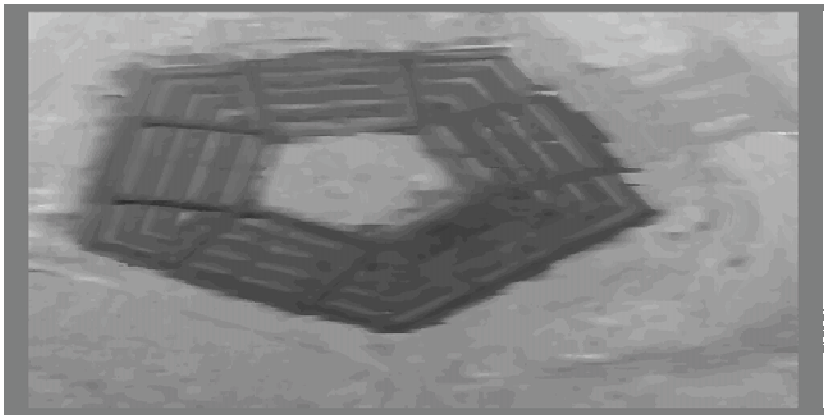
Experiment 2.3.2: Configuring the μ parameter for the illusory square stereo pair from Experiment 1.2. The image below, $\mu = 45$, was chosen as the default value for the μ parameter for the random dot stereo pair since it is the lowest value which strengthens the vertical interaction of the epipolar lines. This value of μ also displays the disparity gradient at the left and right edge of the illusory square. Once again, one can also see that the μ parameter is quite robust, although upon a certain threshold, the vertical interaction becomes too overbearing (such as in the case of $\mu = 100$).

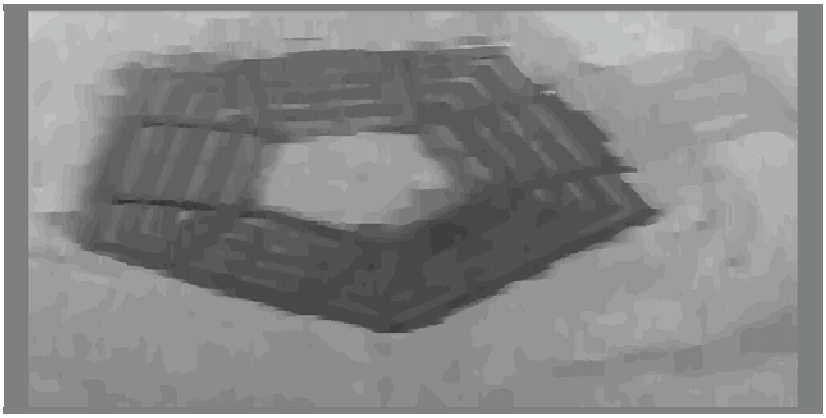


Experiment 2.3.3: Configuring the μ parameter for the pentagon stereo pair from Experiment 1.3. The images below depict the pentagon result for $\mu = 0$ (a), $\mu = 5$ (b), $\mu = 10$ (c). Since this example contains real data, less vertical interaction is

necessary because the data-matching at each epipolar line is quite strong. It is difficult to distinguish these 1024x512 images at the resolution of this paper.

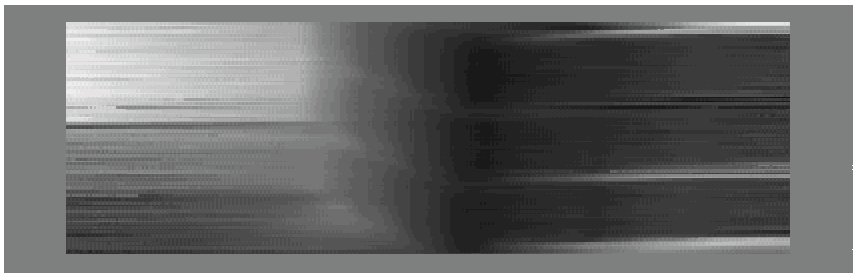
a) 

b) 

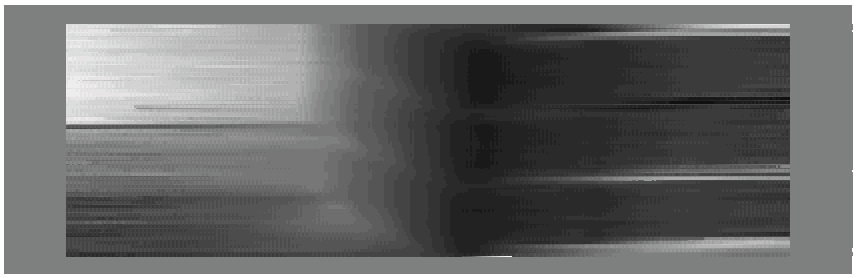
c) 

Experiment 2.3.4: Configuring the μ parameter for the cube stereo pair from Experiment 1.4. The image below, $\mu = 2$, was chosen as the default value for the μ parameter for the cube stereo pair since it is the lowest value which strengthens the vertical interaction of the epipolar lines without blurring the image.

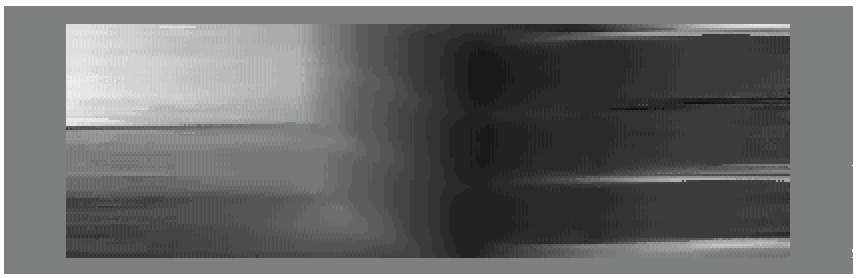
a) $\mu = 0$

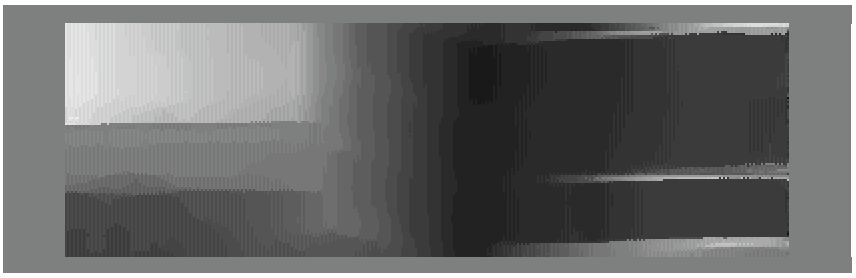


b) $\mu = 2$

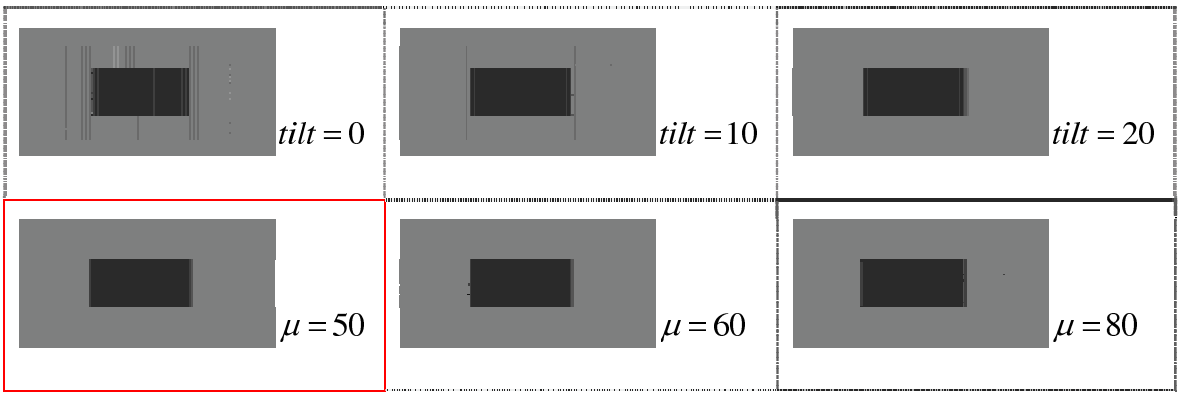


c) $\mu = 5$

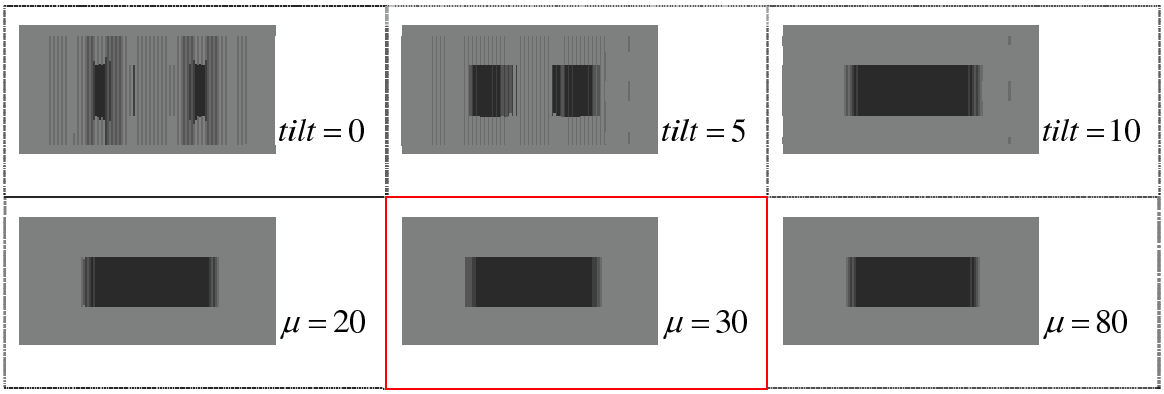


d) $\mu = 10$ 

Experiment 2.4.1: Configuring the *tilt* parameter for the random dot stereo pair from Experiment 1.1. The variations of the *tilt* parameter are displayed below in which the default parameter for this image (red) yields the best disparity solution. This parameter is also quite robust among a variety of values as shown below.

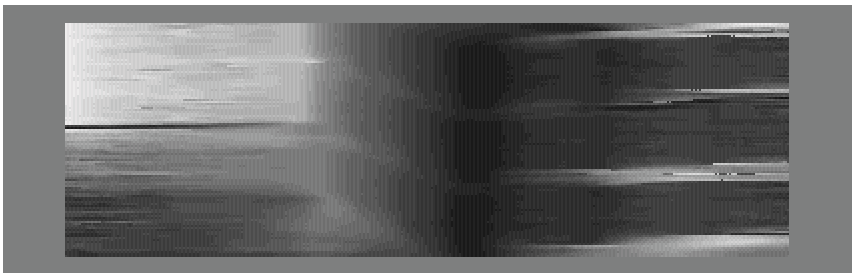


Experiment 2.4.2: Configuring the *tilt* parameter for the illusory square stereo pair from Experiment 1.2. The variations of the *tilt* parameter are displayed below in which the default parameter for this image (red) yields the best disparity solution. This parameter is also quite robust among a variety of values as shown below.

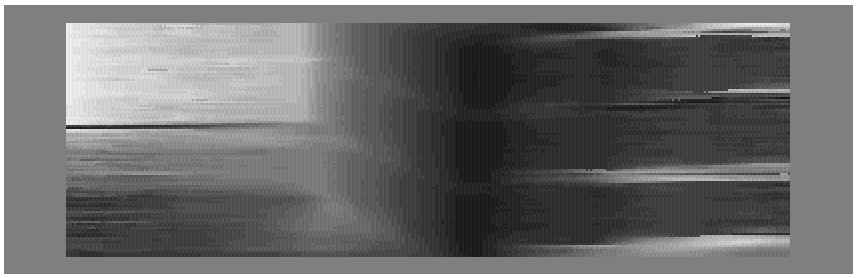


Experiment 2.4.3: Configuring the *tilt* parameter for the cube stereo pair from Experiment 1.4. The variations of the *tilt* parameter are displayed below in which the default parameter for this image $tilt = 60$ yields the best disparity solution. This parameter is also quite robust among a variety of values as shown.

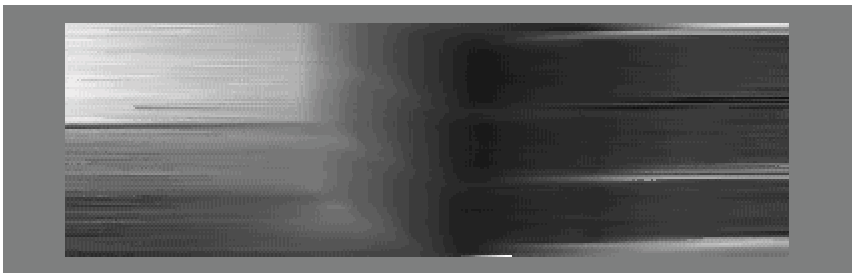
a) $tilt = 0$



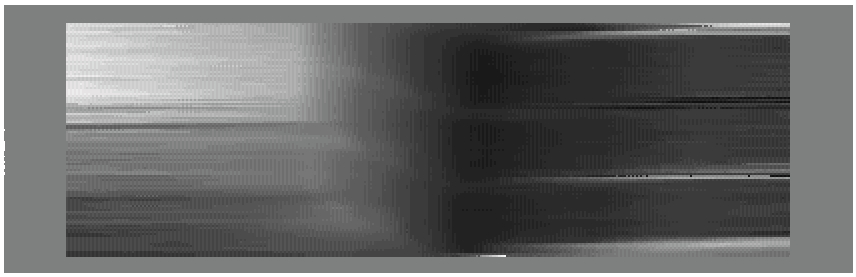
b) $tilt = 10$



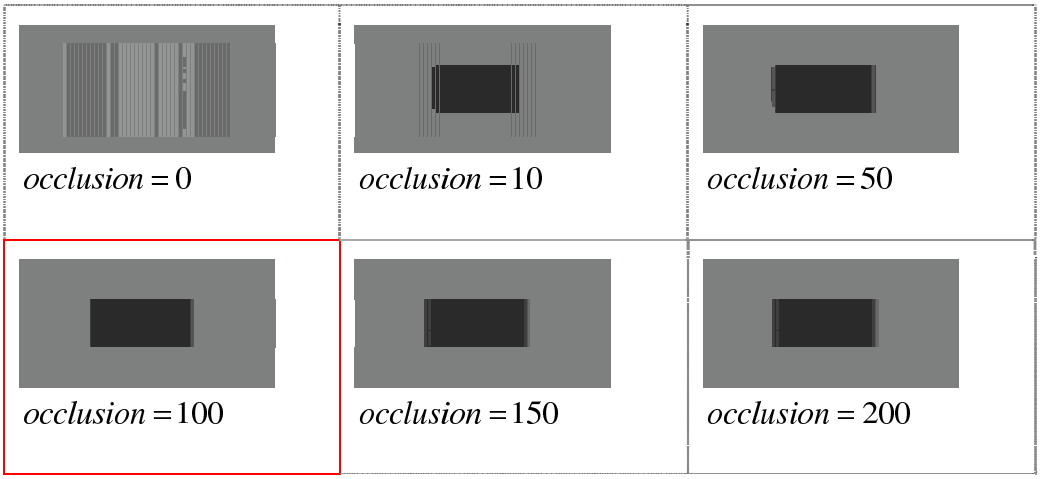
c) $tilt = 60$



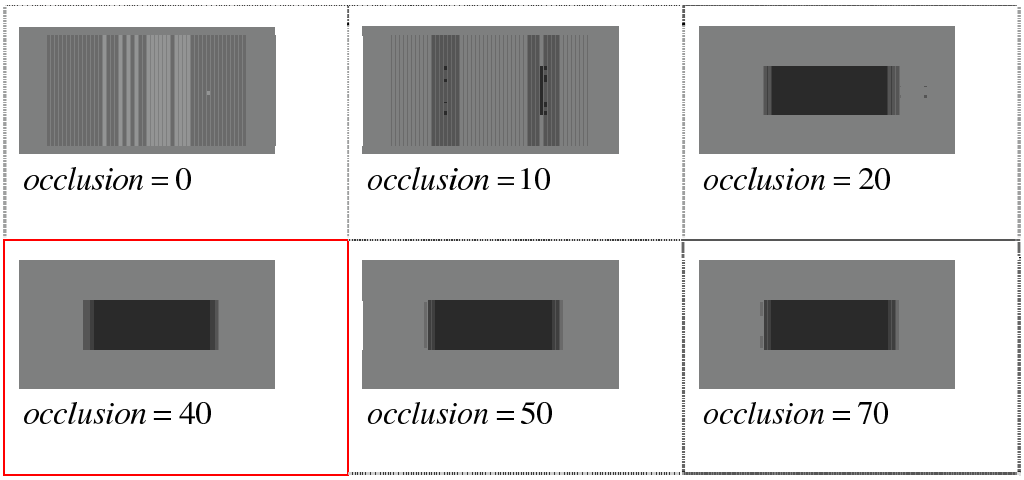
d) $tilt = 100$



Experiment 2.5.1: Configuring the *occlusion* parameter for the random dot stereo pair from Experiment 1.1. The variations of the *occlusion* parameter are displayed below in which the default parameter for this image (red) yields the best disparity solution. This parameter is also quite robust among a variety of values as shown below.

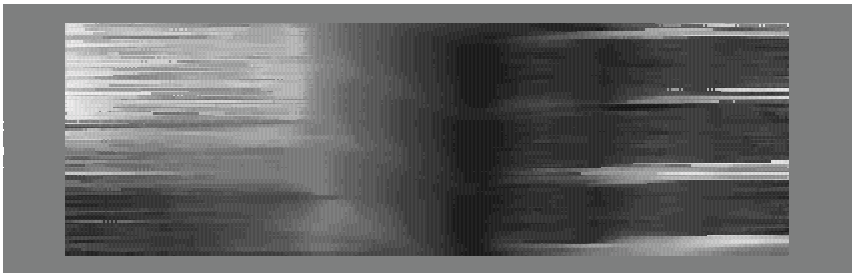


Experiment 2.5.2: Configuring the *occlusion* parameter for the illusory square stereo pair from Experiment 1.2. The variations of the *occlusion* parameter are displayed below in which the default parameter for this image (red) yields the best disparity solution. This parameter is also quite robust among a variety of values as shown below.

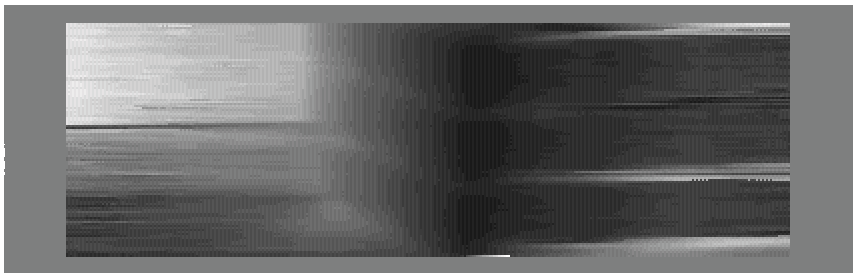


Experiment 2.5.3: Configuring the *occlusion* parameter for the cube stereo pair from Experiment 1.4. The variations of the *occlusion* parameter are displayed below in which the default parameter for this image *occlusion* = 80 yields the best disparity solution. This parameter is also quite robust among a variety of values as shown below.

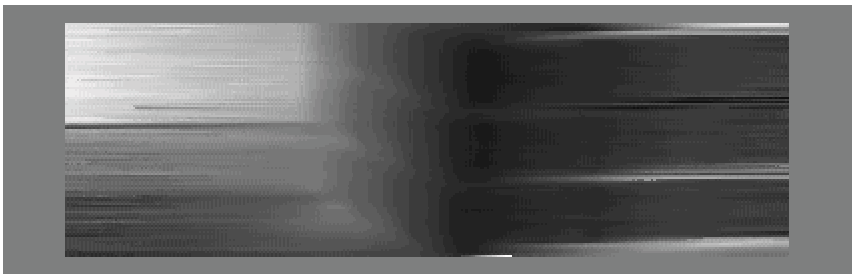
a) *occl* = 10



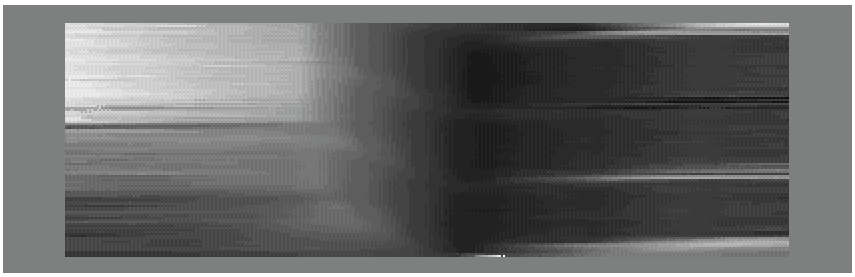
b) $occl = 40$



c) $occl = 80$



d) $occl = 120$



Experiment Round 3: STUMP Algorithm compared with an implementation of the Dynamic Programming Algorithm

Experiment 3.1: Random-Dot Stereogram in which $D = 6$, $\eta = 0$, $tilt = 50, occlusion = 100$, $\mu = 50$, and a scale $s = 3$. This result, using the STUMP

algorithm, is image (a). Image (b) is the same as image (a) with the exception that $\mu = 0$. This illustrates the absence of vertical interaction of epipolar lines. Image (c) is the result of using the dynamic programming algorithm. One can notice that result (a), STUMP with vertical interactions, yields the best solution. It is also worth noting that STUMP with no vertical interaction (b) is still an improvement over the dynamic programming result of (c). This shows that even without the vertical interaction of epipolar lines in STUMP, the belief propagation is favored over the optimal solution performed by dynamic programming.

a) b) c)

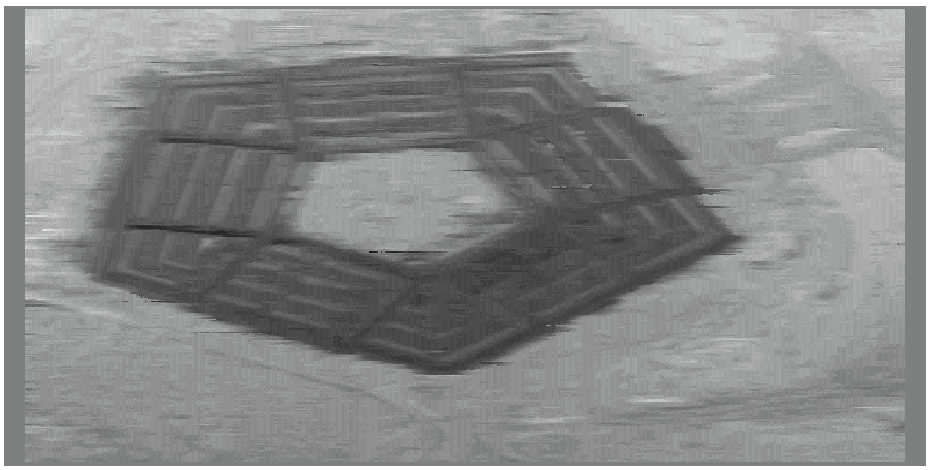
Experiment 3.2: Illusory Square Stereo Pair in which $D = 6$, $\eta = 0.5$, $tilt = 30$, $occlusion = 40$, $\mu = 45$, and a scale $s = 3$. This result, using the STUMP algorithm, is image (a). Image (b) is the same as image (a) with the exception that $\mu = 0$. This illustrates the absence of vertical interaction of epipolar lines. Image (c) is the result of using the dynamic programming algorithm. One can notice that result (a), STUMP with vertical interactions, yields the best solution, and perhaps the only correct solution. It is also worth noting that STUMP with no vertical interaction (b) is still an

improvement over the dynamic programming result of (c), even though the center of the square will never be part of the solution without vertical interactions. The STUMP algorithm can provide a solution to illusory figures where as a dynamic programming (optimal epipolar line) solution cannot.

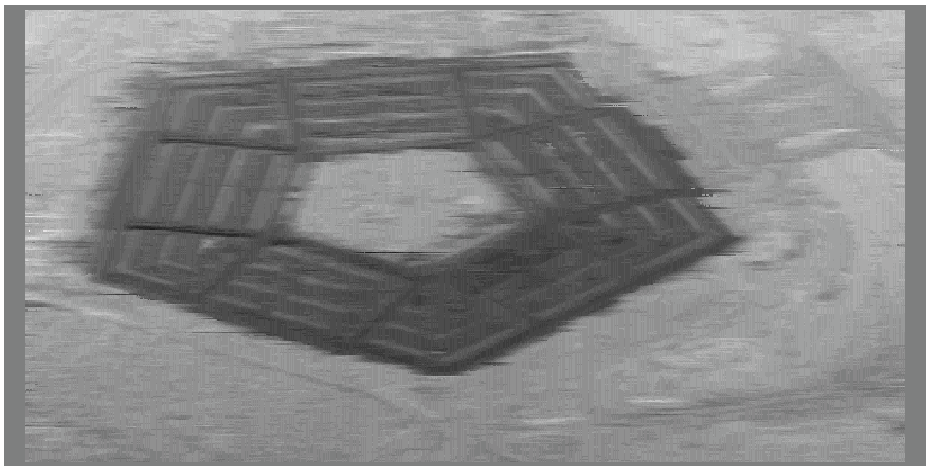
a) b) c)

Experiment 3.3: Pentagon Stereo Pair in which $D = 15$, $\eta = 0.5$, $tilt = 10$, $occlusion = 40$, $\mu = 1$, and a scale $s = 3$. This result, using the STUMP algorithm, is image (a). Image (b) is the same as image (a) with the exception that $\mu = 0$. This illustrates the absence of vertical interaction of epipolar lines. Image (c) is the result of using the dynamic programming algorithm. One can notice that result (a), STUMP with vertical interactions, yields the best solution. It is also worth noting that STUMP with no vertical interaction (b) is still an improvement over the dynamic programming result of (c).

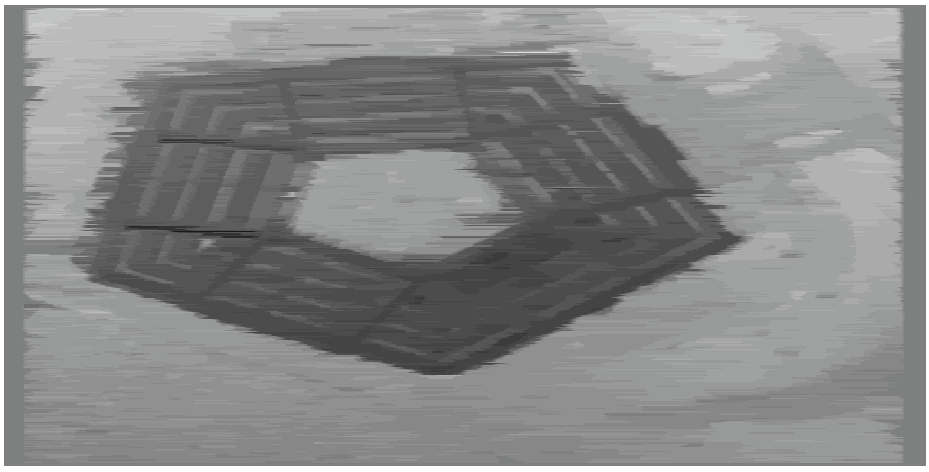
a)



b)

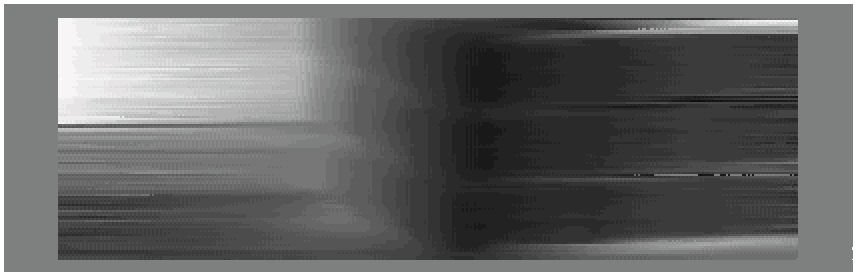


c)

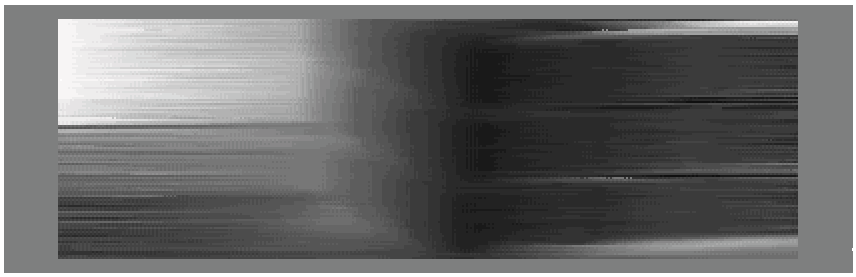


Experiment 3.4: Cube Stereo Pair in which $D = 15$, $\eta = 0.5$, $tilt = 60$, $occlusion = 80$, $\mu = 2$, and a scale $s = 5$. This result, using the STUMP algorithm, is image (a). Image (b) is the same as image (a) with the exception that $\mu = 0$. This illustrates the absence of vertical interaction of epipolar lines. Image (c) is the result of using the dynamic programming algorithm. One can notice that the STUMP algorithm results (a,b), yield a better solution over the dynamic programming (c).

a)



b)



c)

Experiment 3.5: Fruit Stereo Pair in which $D = 25$, $\eta = 0.5$, $tilt = 60$, $occlusion = 60$, $\mu = 2$, and a scale $s = 3$. This result, using the STUMP algorithm, is image (a). Image (b) is the same as image (a) with the exception that $\mu = 0$. This illustrates the absence of vertical interaction of epipolar lines. Image (c) is the result of using the dynamic programming algorithm. One can notice that result (a), STUMP with vertical interactions, yields the best solution. It is difficult to notice the difference of μ at the resolution of this paper. It is also worth noting that STUMP with no vertical interaction (b) is still an improvement over the dynamic programming result of (c).

a)

b)

c)

5 Conclusion

The STUMP algorithm provides an approach to computing disparity from a stereo pair of images by modeling occlusions, discontinuities, and epipolar line interactions. The algorithm solves the correspondence problem in a polynomial time using an estimation of belief propagation in the cyclopean view. We have modeled binocular stereo, including the ordering and smooth surface constraints. We have chosen to solve the problem based on a probability distribution rather than optimal matches along epipolar lines and thus being able to provide a solution for illusory figures. As demonstrated in our results section, the STUMP algorithm provides an accurate estimation of the depth field for stereo images.

Although we are satisfied with the results of our algorithm, there are perhaps other areas to explore. We have considered future ideas regarding a true multi-scale approach of the matching scheme in the cyclopean view. Our future work will include an implementation of a multi-scale pyramid based on the ψ values used in our cost functions to determine the cyclopean disparity map. We also have an interest of exploring more options for using the full probability distribution of disparities. Although one can always imagine other possibilities, we believe that we have made a significant contribution in presenting a solution to the stereo correspondence problem.

Bibliography

[1] Barton Anderson. The role of partial occlusion in stereopsis. *Nature*, 367:365–368, 1994.

[2] Peter N. Belhumeur and David Mumford. A bayesian treatment of the stereo correspondence problem using half-occluded regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1992.

[3] Andrew Blake and Andrew Zisserman. *Visual Recognition.* MIT Press, Cambridge, MA, 1987.

[4] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of mincut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

[5] Peter Burt and Bela Julesz. A disparity gradient limit for binocular fusion. *Science*, 208:615–617, 1980.

[6] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 8(6):679-698, November 1986.

[7] Olivier Faugeras. *Three-Dimensional Computer Vision.* MIT Press, Cambridge, MA, 1993.

[8] Davi Geiger, Bruce Ladendorf and Alan Yuille. Occlusions and binocular stereo. *International Journal of Computer Vision*, 14:211-226, March, 1995.

[9] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[10] William Eric Leifur Grimson. *From Images to Surfaces.* MIT Press, Cambridge, MA, 1981.

[11] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2003.

[12] Hiroshi Ishikawa and Davi Geiger. Occlusions, Discontinuities, and Epipolar Lines in Stereo. In *Proceedings of the Fifth European Conference on Computer Vision*, Freiburg, Germany, June 1998.

[13] Hiroshi Ishikawa and Davi Geiger. Rethinking the Prior Model for Stereo. *Ninth European Conference on Computer Vision (ECCV 2006)*, May 7-13, 2006, Graz, Austria. Lecture Notes in Computer Science 3953, pp. 526-537, Springer-Verlag.

[14] Hiroshi Ishikawa. Multi-scale Feature Selection in Stereo. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, Fort Collins, Colorado, June 1999.

[15] Bela Julesz. *Foundations of Cyclopean Perception.* The University of Chicago Press, Chicago, 1971.

[16] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive window: theory and experiments. In *Proceedings of the Image Understanding Workshop DARPA*, PA, September 1990.

[17] Kai Ju Liu. A Trainable Graph Combination Scheme for Belief Propagation. *PhD dissertation, New York University*. 2006.

[18] Jitendra Malik. On Binocularly viewed occlusion Junctions. In *Fourth European Conference on Computer Vision*, vol.1, pages 167–174, Cambridge, UK, 1996. Springer-Verlag.

[19] David Marr and Tomaso Poggio. Cooperative computation of stereo disparity. *Science*, 194:283–287, 1976.

[20] David Marr and Tomaso Poggio. A computational theory of human stereo vision. In *Proceedings of the Royal Society of London B*, 204:301–328, 1979.

[21] Ken Nakayama and Siunsuke Shimojo. Da vinci stereopsis: depth and subjective occluding contours from unpaired image points. *Vision Research*, 30:1811–1825, 1990.

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

[23] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Francisco, CA, 1988.

[24] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense stereo. *International Journal of Computer Vision*, 47(1):7–42, 2000.

[25] Stephen B Pollard, John E W Mayhew, John P Frisby. Disparity gradients and stereo correspondences. *Perception*, 1987.

[26] Sébastien Roy and Ingemar Cox. A Maximum-Flow Formulation of the N-camera Stereo Correspondence Problem. In *Proceedings of the International Conference on Computer Vision, ICCV'98*, Bombay, India, 1998.

[27] Jian Sun, Nan-Ning Zheng, Heung-Yeung Shum. Stereo Matching Using Belief Propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787-800, July 2003.

[28] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Bethe free energy, kikuchi approximations and belief propagation algorithms. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, Cambridge, MA, 2000. MIT Press.

[29] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In Gerhard Lakemeyer and Bernhard Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*, pages 239–269. Morgan Kaufmann, San Francisco, CA, 2003.

[30] Alan L. Yuille. Cccp algorithms to minimize the bethe and kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14:1691–1722, 2002.

[31] Martin J. Wainwright, Tommi S. Jaakkola and Alan S. Willsky. *MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches. IEEE Transactions on Information Theory*, 51(11):3697-3717, November 2005.

[32] Charles Wheatstone. On some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical Transactions of the Royal Society*, London 128, 371-394, 1838.