STEREO CORRESPONDENCE USING AN ASSISTED DISCRETE COSINE TRANSFORM METHOD

by

Edward Rosales

Bachelor of Engineering, Ryerson University, 2012

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the Program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2015

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Stereo Correspondence using an Assisted Discrete Cosine Transform Method

Master of Applied Science 2015

Edward Rosales

Electrical and Computer Engineering

Ryerson University

Abstract

Many approaches have been taken towards the development of a compliant stereo correspondence algorithm that is capable of producing accurate disparity maps within a short period of time. There has been great progress over the past decade due to the vast increase in optimization techniques. Currently, the most successful algorithms contain explicit assumptions of the real world such as definitive differences in disparity among objects and constant textures within objects.

This thesis starts by giving a brief description of disparity, along with descriptions of some common applications. Next, it explores various methods used in common stereo correspondence algorithms, as well as gives an in depth description and analysis of top performing algorithms. These algorithms are later used to compare with the proposed algorithm.

In the proposed algorithm, frequency stereo correspondence in parallel with the traditional color intensity stereo correspondence is used to develop an initial disparity map. Frequency stereo correspondence is achieved using a winner-take-all block based Discrete Cosine Trans-

form (DCT) to find the largest frequency components as well as their positions to use in disparity estimation. The proposed algorithm uses methods that are computationally inexpensive to reduce the computational time that plagues many of the common stereo correspondence algorithms. The proposed algorithm achieves an average correct disparity rate of 95.3%. This results in a disparity error rate of 4.07% compared to the top performing algorithms in the Middlebury website [1]; the DoubleBP, CoopRegion, AdaptingBP, and ADCensus algorithms that have error rates of 4.19%, 4.41%, 4.23%, and 3.97%, respectively. Additionally, experimental results demonstrate that the proposed algorithm is computationally efficient and significantly reduces the processing time that plagues many of the common stereo correspondence algorithms.

Acknowledgements

First and foremost, I would like to express my sincere and utmost gratitude to my Masters supervisor, Dr. Ling Guan, for his guidance and support throughout my time at Ryerson University. His enthusiasm, inspiration, and advice he had provided me for my Masters was both delightful and productive. He is truly a role model: not only as an enthusiastic and productive researcher, but also someone with a kind heart who cares greatly for others.

It was a blessing for me to be part of the Ryerson Multimedia Research Laboratory (RML). I gained both knowledge and friendship during my tenure at RML, working with my fellow researchers and students. I would like to specifically thank Naimul Khan, Xiaoming Nan, Fei Guo, Ziyang Zhang, Ning Zhang, and Kevin Tang for their friendship and advice they have given me during my time as a Masters student in RML. All of you have made my years in RML one of the most memorable time of my life.

Last but not the least, I would like to thank my family and friends for their unconditional love, encouragement, and continual support for the past two years.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The human visionary system is a complex system that provides the perception of objects in terms of color, texture, motion, and depth [11]. This has been widely explored in stereo vision in the attempt to replicate this complex visionary system. One major problem in computer vision that has become a key topic of research is stereo matching [6]. Stereo matching is defined in [12] as finding the corresponding relationship between pixels from two images taken from the same scene, and to use this correspondence in extracting disparity information. For upcoming applications such as Free-Viewpoint Television (FTV) and Multi-view Television (MTV) stereo matching plays a key role in achieving a high quality experience. Currently, two problems plague many stereo correspondence algorithms which cause them to be unsuitable for real time; (1.) The matching accuracy of the produced disparity map, and (2.) the computational time of the algorithm [9]. Current algorithms are not capable of producing accurate disparity maps while maintaining a fast computation speed. Alternatively, algorithms that are capable of achieving fast computational speeds can only achieve acceptable accuracy rates for small sized images. In the following, we first introduce the definition of disparity as well as different applications and problems associated with stereo correspondence in Chapter 1. Then,

in Chapter 2, we discuss several stereo correspondence techniques as well as several classical stereo correspondence algorithms. Chapter 3 introduces the proposed method that uses frequency components in determining the disparity map. Chapter 4 describes the experimental setup as well as the results that were achieved. Lastly, Chapter 5 concludes this thesis.

1.1 Disparity

To better understand the concept of this work, the definition of disparity should first be explained. First, disparity must be understood in two different instances [13, 14]. Disparity mapping defined in stereo image coding differs from its definition used in stereo vision. Disparity mapping in stereo image coding refers to the representation of the depth information, whereas stereo vision requires the depth information. Here, stereo vision does not necessarily need the true disparity map, the depth information, if the disparity maps corresponding to each camera can be calculated.

It should be understood that schemes such as MTV and FTV all aim to provide the audience with a 3D experience. To properly produce the 3D experience, 3D applications broadcast two separate views, corresponding to the left eye and right eye. Alternatively, a more desirable approach, would be to broadcast one view along with side information. This side information is typically the calculated disparity information of the scene.

1.2 Acquiring Depth Information

There are multiple ways of obtaining depth information from a 3D scene. One example is by using a *laser range camera* and the other is by using a stereo image pair with the assistance of triangulation, this is commonly referred to as *stereo vision, stereo matching* or *stereo corre-*

spondence [15]. In short, *stereo correspondence* refers to the process of matching corresponding pixels from one image to the other. Techniques used to achieve *stereo correspondence* will be explained in further detail in Chapter 2.

1.3 Stereo Correspondence Applications

1.3.1 MTV - Multi-view Television

MTV aims to provide the user with a larger amount of viewing angles though the viewing angle of the scene can not be freely controlled like Free-Viewpoint TV. MTV uses Multi-view Coding (MVC) provided in the H.264/MPEG-4 AVC video compression scheme.

MVC enables efficient encoding of scenes captured simultaneously from multiple cameras. Technically, due to the large amount of data multi-view videos contain, MVC takes advantage of the large amount of inter-view correlation by using efficient predictive coding of neighboring views. A prime example of the implementation of MTV and the MVC coding scheme is the famous *Matrix bullet time* scene where there are a finite number of cameras with a virtual view calculated in between each camera to give the illusion of a moving camera from one point to another.

1.3.2 FTV - Free-viewpoint Television

FTV originally proposed by [5, 16, 17] aims to provide an innovate visual media experience that enables the user to view any 3D scene by freely changing the viewpoint. This application provides the user with the ability to freely control the camera angle and camera location at any point in time.

[5, 18] provides a set of problems that must be resolved for FTV to be realizable. Some of

the more concerning problems are:

- *Representation*: Efficient data representation must be done to best describe all views within a 3D space.
- *Capturing*: Due to the nature of FTV, cameras must be treated differently, where cameras with different characteristics must be treated as a single camera.
- *Rendering*: Since only a finite number of cameras can be used, the remaining infinite number of viewpoints must be generated.

FTV representation uses the Ray-space representation proposed in [19]. The ray-space representation derives a virtual space that incorporates all possible viewing angles through the collection of viewing angles. The ray-space representation will be explained in certain detail in Section 2.3.1, where it produces an infinite number of views within the camera array.

1.4 Fundamental stereo correspondence problems

There are obvious limitations in determining the disparity among a pair of stereoscopic images, though the most apparent of these problems are occlusions, noise and biasing, and the maximum disparity limitations.

1.4.1 Occlusions

There are two types of occlusions present in a pair of stereoscopic images. The first is the occluded regions along the borders of each stereo image. This is caused by the horizontal movement of the camera with respect to the scene. This of course shifts the scene with respect to the horizontal shift of the camera. The second form of occlusions are those present within

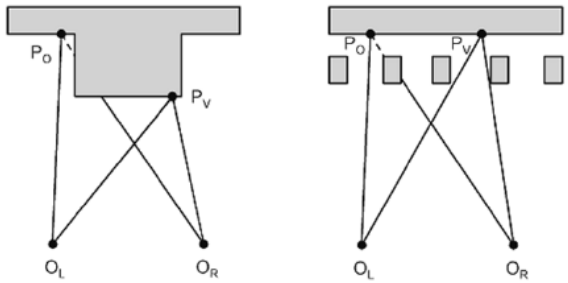

Figure 1.1: Occlusions

the scene when a 3D point in an object is visible in one viewpoint but not the other. The combination of these occlusions can cause multiple problems when trying to achieve an accurate depth map. Figure 1.1 visualizes these two cases of which occlusions can occur. Here, Figure 1.1 shows different scenarios of occlusions on the pixel, P_O.

These occlusions typically happen at an edge of a foreground object where no information from neighboring views are available. Therefore, in stereo correspondence algorithms, these portions of the image are typically left blank or dataless [20]. Since the intensity of occlusions is caused from the shifting distance of one real camera from another, the occlusion intensity can be minimized by reducing the distance between the two cameras.

1.4.2 Noise and Biasing

Modern cameras used in stereo correspondence are typically very susceptible to both noise and biasing. In realistic applications, it is very hard to remove noise and biasing prior to scene capturing, thus to ensure that the lowest matching cost can be achieved between stereoscopic images, both noise and biasing must be done in the preprocessing step through several filters and algorithms.

1.4.3 Maximum Disparity Limitations

Due to the nature of disparity maps and view generation, [21] determined that the horizontal distance between two matching points in a stereoscopic pair of images is limited to a maximum of 3% of the image width. This results in a limitation to the maximum distance between two stereoscopic cameras. Jung et. al. provides an experimental analysis when a pair of stereoscopic cameras are set to produce disparity values larger than 3%. Here, [21] states that many users began to feel large amounts of eyestrain when a virtual view was generated between the two stereoscopic images.

1.5 Contributions and Motivations

The sizable evaluation database in [1] shows that most, if not all, stereo correspondence algorithms solved initial disparity calculations using color intensity comparisons among four stereoscopic pairs. Thus a motivating factor for this paper is to approach the stereo correspondence problem using a different method that may potentially provide more accurate results in higher detailed regions while achieving faster computational times.

The proposed method focuses on the adaptation of frequency based features for stereo based matching [10]. The proposed algorithm searches for the top frequency components of each segment, which are then taken and compared across a database as an indicator as to which segment in the database matches the input segment. Typical stereo correspondence algorithms such as [6], [7],[8], and [9] use color intensity comparisons to determine the preliminary disparity map whereas the proposed algorithm uses top magnitude frequency components for the same task. Thus, the proposed method aims to diverge from the common methods of stereo correspondence by proposing a preliminary method that uses local frequency methods to deter-

mine the disparity map. This is in hope of removing any computationally expensive methods such as image segmentation and other iterative algorithms.

The proposed algorithm was initially motivated by the performance that was achieved in [10]. Here, Wang achieves a near 100% recognition rate in audio files by sampling the target audio files at the maximum frequency points. The algorithm samples the top 30 frequency magnitudes and positions per second thus providing a large database comparisons. Additionally, the performance achieved in [22, 23] accompany the feasibility of using frequency components in determining the frequency components of a stereoscopic pair of images.

Chapter 2

Literature Review

2.1 Introduction

This chapter focuses on different applications, approaches, and considerations needed for stereoscopic imaging. Here, camera calibration, along with the accompanying methods behind camera calibration must be taken into consideration. Due to the setup of cameras for stereoscopic imaging, camera calibration must be done for every configuration. Along with this, the theory behind matching pixels will be briefly explained in Section 2.2.2.

After camera calibration is presented, the conversion from disparity to depth is explained. Here, the usefulness of disparity maps can be seen, as the derived disparity map can provide accurate depth information used for applications such as 3DTV, MTV, and FTV.

This Chapter will conclude by going into a detailed explanation of the top performing algorithms found in [1]. Here, each of these algorithms use at least one cost function explained in Section 3.3 for finding the disparity map.

2.2 Camera Calibration

Due to the arrangement of cameras in a stereo vision system, there will always be misalignment among multiple variables that must be taken care of to ensure the process of finding matching points can be achieved in the easiest manner, therefore camera calibration is an essential step in stereo correspondence. [24] proposes a camera calibration technique that observes a planar pattern in a number of different orientations. If it is assumed that the model plane of the world coordinate system is located at Z=0, then the relationship between a real world point, M, and its image projection, m, can be written to resemble that in Equation 2.2.

$$s\tilde{m} = A \begin{bmatrix} R & t \end{bmatrix} \tilde{M} \text{ with } A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{2.2}$$

For further understanding of camera calibration, the constraints held on the intrinsic parameters of a camera need to be first defined. If the homography of a camera can be defined by 2.3, then 2.4 can be realized if it is assumed that the homography is effected by some arbitrary constant, λ.

$$s\tilde{m} = H\tilde{M} \text{ with } H = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \tag{2.3}$$

$$\begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \tag{2.4}$$

For those interested in looking more into other methods for camera calibration should look at [25], [26], and [24] where each citation discusses different methods for camera calibration for stereoscopic pairs.

2.2.1 Image Rectification

In non-controlled environments, it is almost impossible to keep cameras perfectly aligned with each other with no rotation. Additionally, camera distortion causes a mismatch in pixel locations with respect to their actual location. To relieve this problem, rectification is performed and is actually two-fold beneficial in the computation of stereo correspondence. First, rectification significantly reduces the computational complexity of the pixel matching algorithm. Secondly, by fitting each image onto a mutual plane, pixel matching can be done in one direction in comparison. Figure 2.1 helps provide a clearer description of image rectification. As seen in Figure 2.1, image rectification provides each image in a stereo pair to be fitted into one mutual plane. The dotted blue lines shown in Figure 2.1 are known as epipolar lines.

2.2.2 Epipolar Lines

Typically, a multitude of cameras is needed to achieve stereo vision. Though in Section 2.2, camera calibration was briefly introduced, the concept to achieve stereo vision can be summed to the derivation of epipolar lines. Epipolar lines, as shown in Figure 2.2 represents the relationship between two neighbouring cameras. From this point, Figure 2.2 will be referenced to describe the epipolar equation.

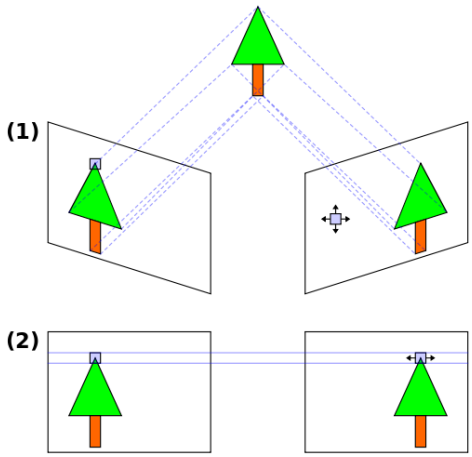

Figure 2.1: a) Unrectified image. b) rectified image

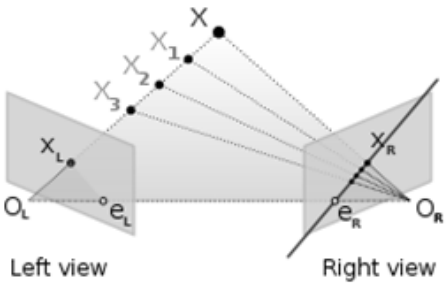

Figure 2.2: Epipolar lines

If O_L and O_R are considered as the two camera centers, than X_L and X_R are the two intersection points of the projected rays on the image plane on the left and right cameras, respectively. It is inferred that any 3D point in the real world, X_1, X_2, or X_3, lies on the projecting line of $O_L X$, the 2D projection of real world coordinates onto the left camera image plane. Similarly, any of these points lie on a different location defined by the line $O_R X_x$, where X_x is defined by the location of the point in the real world. However, rather then deriving epipolar lines for each unique real world point, each point in the right image can be derived from $O_L X_L$. Prior to deriving the epipolar equation, it should also be stated that any real world point lying on the $O_L X$ line will have the right image projection of said point within the range of O_R and e_R. Similarly, any point lying on $O_R X_R$ will be projected to the left image within the range of O_L and e_L, these relationships between the left and right image are the so-called epipolar lines, which contain all the projection points from the principle ray of another view point.

Referring back to the derivation of the epipolar equation, [27] and [28] provide an in-depth derivation for the epipolar equation, here it can be assumed that the transformation equations of the two cameras can be defined as that in Equation 2.5.

$$\begin{aligned} Z_{c1} u_1 &= M_1 X = [\ M_{11} \quad m_1\ ] X \\ Z_{c2} u_2 &= M_2 X = [\ M_{21} \quad m_2\ ] X \end{aligned} \tag{2.5}$$

Where $X=[X_W\ Y_W\ Z_W\ 1]^T$ is the homogeneous real world coordinate point; u_1 and u_2 represent the corresponding image points to the real world point X, M_{11} and M_{21} are the rotation matrices of each camera, m_1 and m_2 are the translation matrices of each camera, and Z_{c1} and Z_{c2} are the scalar factors in the image domain. Defining the 3D coordinate system of X as $x=[X_W\ Y_W\ Z_W]^T$, Equation 2.5 can be expanded to:

$$\begin{aligned} Z_{c1}u_1 &= M_{11}x + m_1 \\ Z_{c2}u_2 &= M_{21}x + m_2 \end{aligned} \tag{2.6}$$

Canceling x, equation 2.7 yields:

$$Z_{c2}u_2 - Z_{c1}M_{21}M_{11}^{-1}u_1 = m_2 - M_{21}M_{11}^{-1}m_1 \tag{2.7}$$

The right hand side of Equation 2.7 defines a vector that corresponds to an inverse symmetric matrix, therefore assuming $m = m_2 - M_{21}M_{11}^{-1}m_1$ and the corresponding matrix is m_x, Equation 2.7 can be written as:

$$u_2^T m_x M_{21} M_{11}^{-1} u_1 = 0 \tag{2.8}$$

Which denotes the epipolar equation. Looking at Equation 2.8, it can be seen that assuming the image points, u_1 and u_2 are given, the outcome of the equation is determined solely on the transform matrices, M_1 and M_2, of the two cameras. This results in a new term that defines the relationship among the two rotation matrices of the cameras. This term is typically defined as the Principle Matrix between cameras and can be defined in Equation 2.9.

$$F = m_x M_{21} M_{11}^{-1} \tag{2.9}$$

2.3 Real World Acquisition and Stereoscopic Images

For real world applications, real time video capture of the scene is needed. [29] uses a 100 camera system, although the camera setups can range from a dense configuration [2], as shown in Figure 2.3, to an intermediate camera configuration [3], as shown in Figure 2.4, to a wide

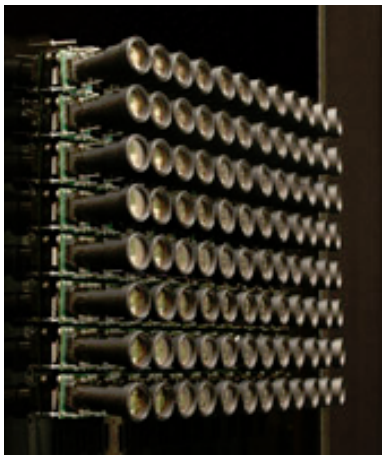

Figure 2.3: Dense Camera Configuration [2]

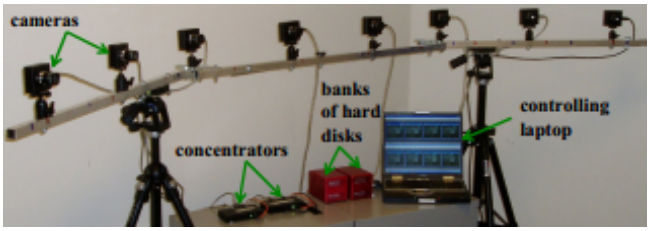

Figure 2.4: Intermediate Camera Configuration [3]

camera distribution [4], as shown in Figure 2.5. As explained in [30], several aspects come into determining which camera configuration, as well as which camera would best fit the target application. A dense camera configuration allows effects such as synthetic aperture and focusing [31], though due to the amount of cameras needed for the dense camera configuration, there is a large number of images needed for rendering. Similarly, the wide camera distribution is the only configuration among the discussed configurations that allow a full 360 degree range of viewing angles, though due to the separation distance of each camera, occlusions become a much more apparent problem than that of the other configurations.

Unlike the images that are used for testing, real world images [3] are much more difficult to deal with, mainly due to the camera placement in uncontrolled environments and the rec-

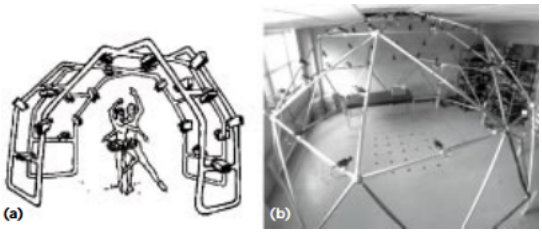

Figure 2.5: Wide Camera Configuration [4]

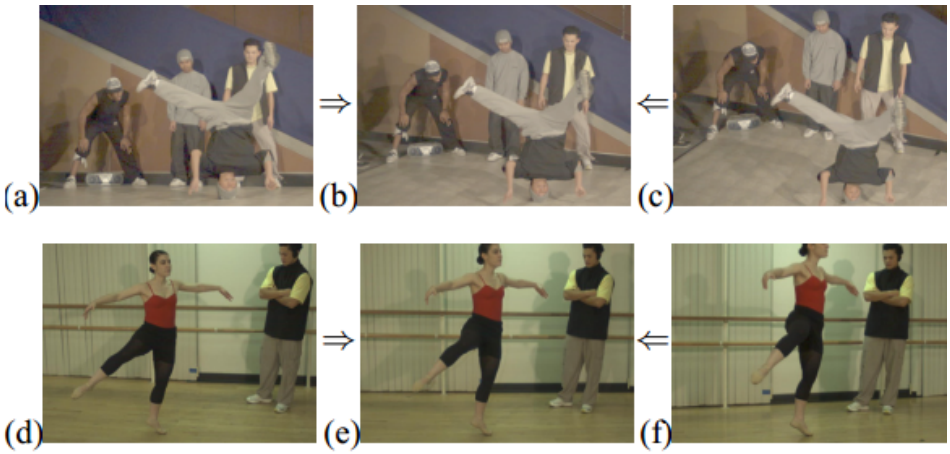

Figure 2.6: Real World Images [3]. (a) and (c) are real world images of the breakdancers image sequences and (b) is a virtual view. Similarly, (d) and (f) are real world images of the ballerina image sequence and (e) is a virtual view.

tification that has to be done in real-time. [3] present a set of images that were captured in a semi-controlled environment. Here, semi-controlled environment is used because the camera placement in the scene is predetermined and the camera array in Figure 2.4 follows the rules set in place by the limitations defined in [21]. Figure 2.6 provides an example of the real world image sequences, *breakdancers* and *ballerina* provided by [3].

2.3.1 Ray-Space Representation

The Ray-Space representation mainly used in FTV allows the user to view any 3D scene from an infinite amount of views, of course, other methods to produce these views are possible as

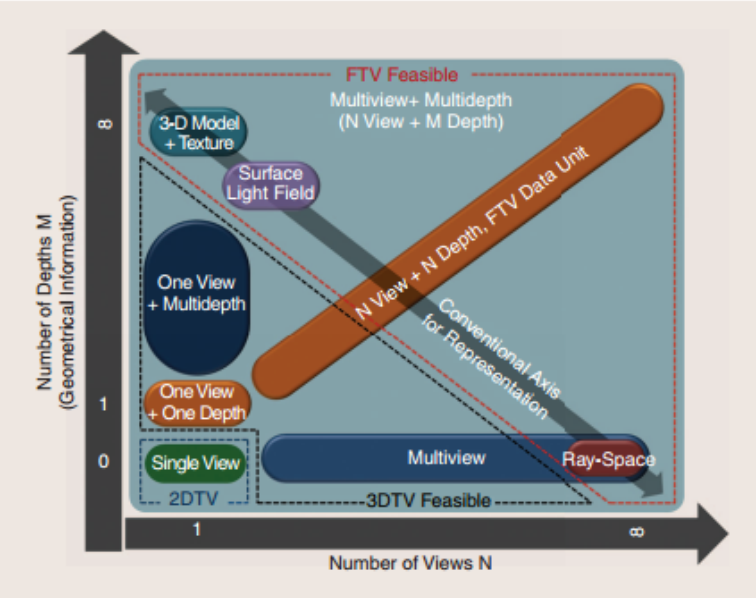

Figure 2.7: View geometry representation[5]

seen in Figure 2.7, though the most common method used for view generation is the Ray-space representation.

There are two typical forms of ray-space that are used in FTV applications; the orthogonal ray-space representation, and the spherical ray-space representation. Here, the images captured in the real world plane are converted to the corresponding ray-space domain, where transformed images are aligned in parallel. These parallel slices form the 3D environment which can then be sliced in any direction to obtain the corresponding view. For those wanting to understand the more about the ray-space representation can look into [5] and [19].

2.4 Survey of Classical Stereo Correspondence Algorithms

There are many different algorithms in [1] that attempt to solve the problems that persist in depth estimation. There are currently over 140 stereo correspondence algorithms in the mid-

dlebury database. The Double Belief Propagation (DoubleBP) [6], CoopRegion [7], Adapting Belief Propagation (AdaptingBP) [8], and the ADCensus [9] algorithms are some of the top performing algorithms.

2.4.1 DoubleBP

The DoubleBP algorithm proposed by Q. Yang et al. [6] uses an iterative refinement module based on a weighted color correlation scheme to achieve a confident initial disparity map. The DoubleBP algorithm can be simplified to three separate modules; (1) the initialization module, (2) the pixel classification module, and (3) the iterative refinement module. The initialization module first determined the correlation volume for both the left and right images based on the color-weighted correlation. The color-weighted correlation that is used in [6] is defined as the absolute difference of luminance levels between two images , though it is mentioned that other methods for the volume correlation construction can be used. Equation 2.10 shows the color difference between pixels x and y in the color channel C. Next, the weight of each pixel is found within the support window of each other corresponding pixel, as seen in Equation 2.11. Here, [6] defines $\beta_{cw} = 10$ and $\gamma_{cw} = 21$ which were defined empirically through experimentation.

$$\triangle xy = |I_c(x) - I_c(y)|/3 \tag{2.10}$$

$$\omega_{xy} = e^{-(\beta_{cw}^{-1} \triangle_{xy} + \gamma_{cw}^{-1} ||x-y||_2)} \tag{2.11}$$

Next, the correlation volume matrices of each image are found usign the support window of each pixel and the Birchfield and Tomasi pixel difference, defined in Equation 2.12, where W_x is the support window in the x axis, $d(y_L, y_R)$ is the Birchfield and Tomassi pixel difference.

Along with the initial disparity maps, D_L^0 and D_R^0, the initial data term is also outputted from the initialization module. The initial data term is a linear transform based on the correlation volumes, where the maximum of the correlation volume and a preset volume is taken for each pixel.

$$C_{L,x_L}(d_x) = \frac{\sum_{(y_L,y_R)\in W_{X_R} x W_{X_R}} \omega_{W_{X_L y_L}} \omega_{W_{X_R y_R}} d(y_L, y_R)}{\sum_{(y_L,y_R)\in W_{X_R} x W_{X_R}} \omega_{W_{X_L y_L}} \omega_{W_{X_R y_R}}} \tag{2.12}$$

The pixel classification module classifies each pixel as one of three possible labels: *occluded, stable* and *unstable*. A pixel is defined as occluded if the mutual consistency check defined in Equation 2.13 does not pass, where D_R and D_L are the right and left disparity maps, respectively. In order to determine whether a pixel is label as stable or unstable, a correlation confidence check is performed. Equation 2.14 defines the correlation confidence between the cost of disparity of the first iteration, C_L^1, and the cost of disparity of the second iteration, C_L^2. τ_1 defines the preliminary predetermined threshold value needed to defined a pixel as unstable, where τ_2 defines the predetermined threshold needed to achieve a stable classification.

$$D_L(x_L) = D_R(x_L - D_L(x_L)) \tag{2.13}$$

$$\tau_1 < |\frac{C_L^1 - C_L^2}{C_L^2}| < \tau_2 \tag{2.14}$$

The Iterative Refinement module propagates the stable pixels onto the unstable and occluded pixels using the hierarchical belief propagation method. This is done by using the main building blocks of the iterative module, these blocks consist of mean shift color segmentation, plane fitting, data term formulation, and a hierarchical belief propagation process. The mean shift color segmentation is also performed on the image. Similarly, plane fitting is done by

applying the RANSAC method on stable pixels found through the pixel classification module. The data term formulation is determined from the output of the plane fitting algorithm, D_{pf}. To regularize the estimation process of stable, nonstable, and occluded regions for the date term formulation, Equation 2.15 is first applied, where $D_L^{(i+1)}$ is the disparity map of the left image after the $(i^{th}+1$ iteration, and $D_{pf}^{(i)}$ is the RANSAC output after the i^{th} iteration. The data term formulation is then defined differently according to the label of each pixel, as defined in 2.16, where k_o, k_u, and k_s represent the regularization constant needed for data term regularization. [6] defines each constant as 2.0, 0.5, and 0.05 respectively for occluded, unstable, and stable constants to reflect the fact that occluded and stable regions require the most regularization. Lastly, after each iteration, belief propagation is done to achieve a more stable disparity map.

$$a_i = |D_L^{(i+1)} - D_{pf}^{(i)}| \tag{2.15}$$

$$E_D^{i+1} = \begin{cases} k_o a_i, & \text{occluded} \\ E_D^{(0)} + k_u a_i, & \text{unstable} \\ E_D^{(0)} + k_s ai, & \text{stable} \end{cases} \tag{2.16}$$

The module iterates itself while updating the disparity and data terms until a confident disparity map is obtained. Figure 2.8a shows the resulting depth map achieved after depth enhancement of multiple test sets, whereas Figure 2.8b shows the ground truth of each test image respectively.

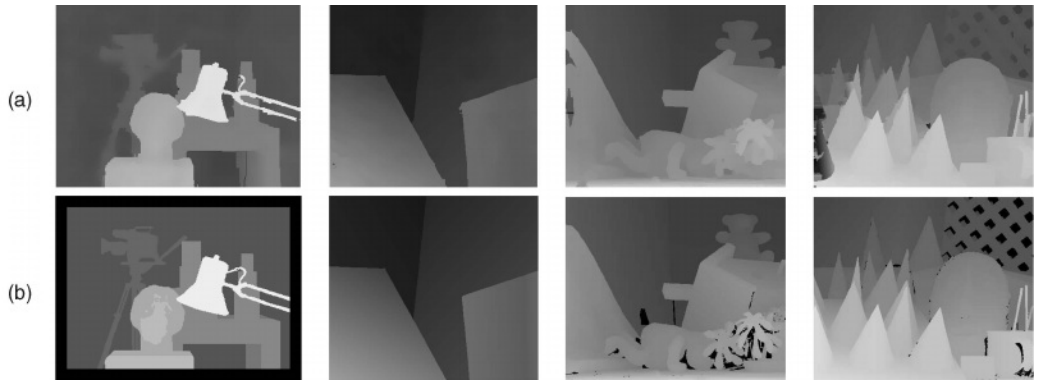

Figure 2.8: a) DoubleBP results and b) ground truths [6]

2.4.2 CoopRegion

The CoopRegion proposed by Z. Wang and Z. Zheng [7] achieves an accurate representation of the disparity map using an image segmentation algorithm and an adaptive correlation method. The CoopRegion algorithm uses the Mean-Shift algorithm to segment the left image of the stereo pair. Once the left image is segmented, a stereo matching algorithm is employed. In [7] a Winner-Take-All strategy is combined with the adaptive correlation window represented in [32] is used to achieve the initial disparity map of the stereo pair. In order to achieve a more accurate disparity map, a voting based plane fitting algorithm was developed. The plane fitting algorithm uses the matching reliability of each pixel to determine the direction of the disparity plane. After the plane fitting algorithm, some outliers may still be present, thus to remove the remaining outliers, the RANSAC algorithm presented in [33] is used. Lastly, in order to achieve an optimized disparity map, a cooperative optimization technique divides a region into several subregions and optimization is performed on each suberegion looking at each corresponding energy functional as shown in Equation 2.17, where E_i is the energy functional for sub-region i, λ_i is the influence of the target subregion on the entire region, ω_{ij} is the corresponding weight of adjacent region j to target subregion i, and k represents the k^{th} iteration of the subregion. Thus

by reducing the total energy function in Equation 2.17 by minimizing the energy functions in Equation 2.17, a reasonable disparity map can be obtained, iterating through this step until the total energy function converges.

$$\psi_i^k(x) = (1-\lambda_i)E_i(x) + \sigma_i \sum_{j \neq i} \omega_{ij} E_x(x) \quad i, j = 1...n \tag{2.17}$$

$$E_i = E_{data} + E_{occlude} + E_{smooth} \tag{2.18}$$

The data cost is computed by looking at direct pixel-wise matching where a penalty cost is applied depending on the label of the pixel. the occlusion cost is computed by the projection of pixels as shown in equation 2.20, similar to that of the mutual consistency check described in the DoubleBP algorithm, where λ_{occ} is the penalty constant of an occluded pixel. Once the occluded energy for each pixel is calculated, the total energy is found through Equation 2.20, where $|Occ_L|$ and $|Occ_R|$ are the total number of left and right occluded pixels, respectively. The smoothness cost is only added when a difference between two neighboring pixels with different disparity levels are found, some examples of this are the borders of objects within the image. Equation 2.21 shows the smoothness energy function, where λ_s is the smoothness penalty constant, and B_c is the border pixels of the target region.

$$E_occludeq = \begin{cases} \lambda_{occ}, & \text{if q is a lect occlusion pixel} \\ \lambda_{occ}, & \text{if q is a right occlusion pixel} \\ 0, & \text{Otherwise} \end{cases} \tag{2.19}$$

$$E_{occlude} = (|Occ_L + |Occ_R|)\lambda_{occ} \tag{2.20}$$

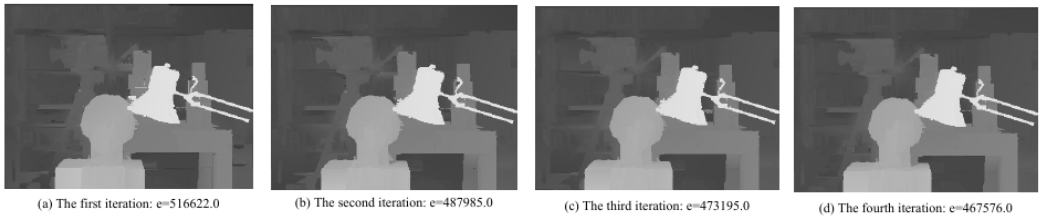

Figure 2.9: Disparity map through four iteration using CoopRegion [7]

$$E_{smooth} = \sum_{p \varepsilon B_c} \begin{cases} \lambda_s, & \text{if } |d(p) - d(q)| \geq 1 \\ 0, & \text{Otherwise} \end{cases} \tag{2.21}$$

Figure 2.9 shows the progress of the Tsukuba disparity map after four iteration of the cooperative optimization algorithm, as it can clearly be seen, the distinction between each segment becomes more clear after each iteration.

2.4.3 AdaptingBP

The AdaptingBP algorithm proposed in A. Klaus et. al [8] applies a combination of techniques discussed in previous algorithms to achieve a more accurate disparity map. Similar to that of the CoopRegion, mean-shift color segmentation is first applied, though in the AdaptingBP algorithm, the mean0shift segmentation [34] is applied to both images in the stereo input. Since the total amount of segments is unknown, its is in best practice to perform over-segmentation, at which point unnecessary segments will be removed in later steps. The next step is to perform local matching on the segmented stereo pair. Typically, one local matching dissimilarity measure is used to achieve the disparity planes, but in the case of AdaptingBP a combination of two local matching dissimilarity measures are used; A sum of absolute intensity difference SAD is used in combination with a gradient based measure were used to perform self-adapting dissimilarity measure that would outperform any single dissimilarity measure by making the

combination take the advantages of each difference measure such as the robustness to change in camera gain and non-lambertian surfaces at the cost of a low discriminating power [8] obtained from the gradient based dissimilarity measure. Equations 3.4 and 3.3 define the SAD and gradient based dissimilarity measures respectively, where I_R is the right image, I_L is the left image, d is the disparity level, and N is the surrounding window of the target pixel. In order to combine each of the dissimilarity measures, an optimal weight, ω, must be determined by performing a correlation confidence check to maximize the number of reliable correspondences that are filtered out. Equation 2.22 defines the final cost function that is used for the algorithm.

$$C(x,y,d) = (1-\omega) * C_{SAD}(x,y,d) + \omega * C_{GRAD}(x,y,d) \tag{2.22}$$

Once the disparity planes are found using the reliable correspondences derived from Equation 2.22, a robust plane fitting algorithm is performed to ensure that a reliable depth map is achieved. The proposed plane fitting algorithms outperforms a decomposition method to solve for the parameters, shown in Equation 2.23 , for each disparity plane, where a, b, and c are plane parameters.

$$d = a * x + b * y + c \tag{2.23}$$

The proposed method, first, estimates the horizontal slant using all reliable disparities lying on the same horizontal line of each segment. The derivative of the disparity planes over all x values is then used to determine the horizontal slant using the convolution of a Gaussian Kernel. Similarly, the vertical slant is estimated in the same method as that of the horizontal slant. Once both slants are found, the center disparity of the segment is estimated. Lastly, in order to optimize the disparity map, the prior steps are iterated to minimize the energy function in Equation 2.18. Figure 2.10 shows the obtained disparity maps of the Tsukuba and Venus

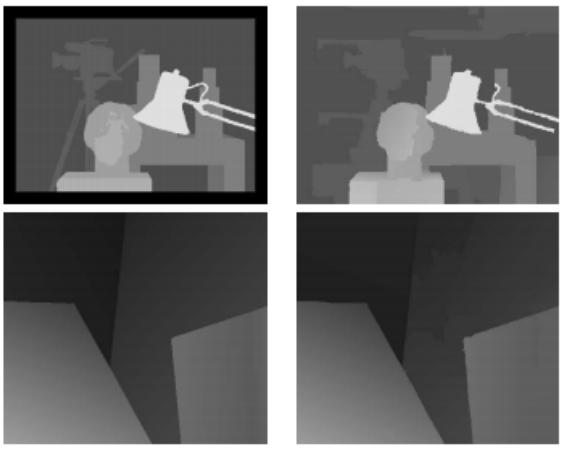

Figure 2.10: Ground truh and generated disparity maps for Tsukuba and Venus using AdaptingBP [8]

databases alongside their respective ground truths when using the AdaptingBP algorithm.

2.4.4 ADCensus

The ADCensus algorithm proposed by X. Mei et al. [9] approaches the disparity estimation problem from a different direction than those earlier discussed. The proposed algorithm first determines the initial cost of disparity calculation based on a combination of preliminary cost functions. The first preliminary function is the SAD function defined in Equation 3.4 and the second cost function if found through the census transform. The census transform encodes each pixel with a bit string relative to its surrounding pixels. This transform reduces variation effects experienced in cameras such as gain and bias as well as making the resulting image more tolerable to potential outliers and image noise. This transformation was first proposed in [26] where the *rank* and *census* transforms are defined. The *rank* transform is defined as a non-parametric measure of local intensity. It is a measure of the number of pixels in the local region whose intensities are lower than that of the target pixel. Similarly, the *census* transform defined in [26] maps the local region of the target pixel to a bit string which represents the pixels in the

local region where pixel intensities were lower than the target pixel. Although both methods are non-parametric and achieve a measure based on pixel intensities along the local region of the target pixel, the benefits of the *census* transform overshadow those of the *rank* transform. However, this transform also produces ambiguities in image regions with repetitive or similar structures. Thus, the SAD and *census* cost functions form a final cost function defined in Equation 2.24 and 2.25 where p is the target pixel, and the parameter λ controls the influence of outliers on the final cost.

$$C(\rho,d) = \rho(C_{Census}(\rho,d),\lambda_{Census}) + \rho(C_{SAD}(\rho,d),\lambda_{SAD}) \tag{2.24}$$

$$\rho(c,\lambda) = 1 - exp(-\frac{c}{\lambda}) \tag{2.25}$$

As mentioned in [9], the purpose of this combination is twofold. Firstly, using the function ρ maps both cost functions to a total range between 0 and 1, such that the outcome isn't severely biased by one cost function. Also, with the use of parameter λ, easy control of influence of parameters is possible for a wide range of stereo pairs. [9] shows the preliminary disparity results achieved when using this ADCensus cost function, it can clearly be seen that improvements are achieved for both repetitive structures and textureless regions. The next step done for the proposed algorithm is cost aggregation to reduce the ambiguities and noise in the image. The method proposed by Zhang et al. [35] uses a 2-dimensional aggregation method and a constructed upright cross in determining the new cost of the target pixel. Unlike [35], X. Mei et al. produced an enhanced set of rules in determining the upright cross. Assuming p is the target pixel, p_1 is an endpoint pixel along the arm, τ_1 and τ_2 are intensity variations where $\tau_2 < \tau_1$, and L_1 and L_2 are arm lengths, where $L_2 < L_1$.

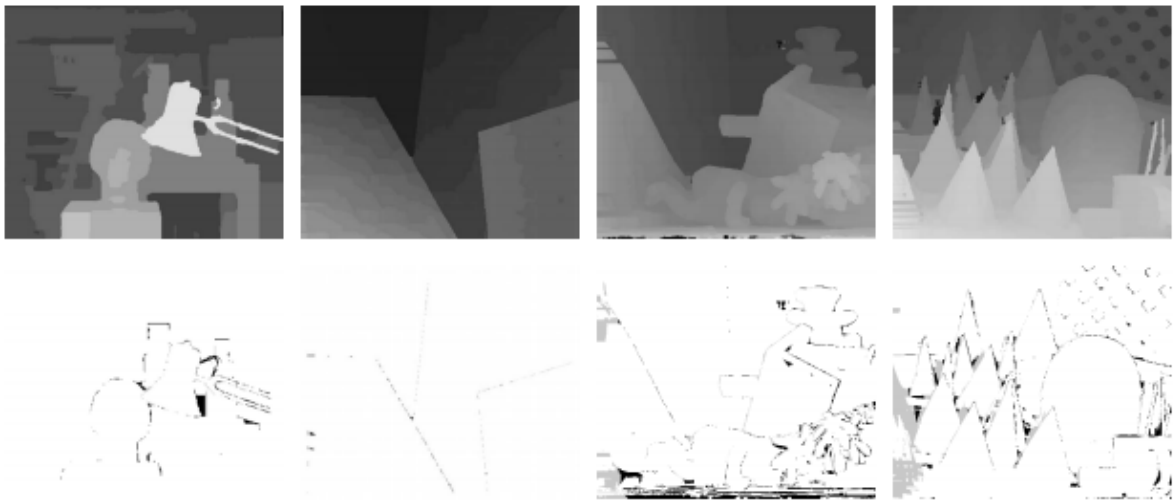

Figure 2.11: Ground truth and generated disparity maps using ADCensus [9]

1. $\mathrm{D}_C(p_1,p)¡\tau_1 and D_C(p_1,p_1+(1,0))¡\tau_1$

2. $\mathrm{D}_S(p_1,p)¡L_1$

3. $\mathrm{D}_C(p_1,p)¡\tau_2, if D_S(p_1,p)¡L_1$

The three enhanced rules are placed when performing cost aggregation to ensure that the color between pixels are similar, and to allow more flexibility in the production of the arm lengths, This cost aggregation algorithm is iterated four times to ensure that stable cost values are obtained. Lastly, scanline optimization presented by Hirschmuller's semi-global matching method [36] and a multi-step disparity refinement step is done to reduce the effects of outliers and discontinuities present in the disparity map. Figure 2.11 shows the generated disparity maps for the Tsukuba, Venus, Teddy, and Cones stereo pairs provided by the Middlebury database [1] along with the errors when compared to the ground truth.

Algorithm	Tsukuba			Venus			Teddy			Cones			avg. % of
	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	bad pixels
DoubleBP	0.88	1.29	4.76	0.13	0.45	1.87	3.53	8.30	9.63	2.90	8.78	7.79	4.19
CoopRegion	0.87	1.16	4.61	0.11	0.21	1.54	5.16	8.31	13.0	2.79	7.18	8.01	4.41
AdaptingBP	1.11	1.37	5.79	0.10	0.21	1.44	4.22	7.06	11.8	2.48	7.92	7.32	4.23
ADCensus	1.07	1.48	5.73	0.09	0.25	1.15	4.10	6.22	10.9	2.42	7.25	6.95	3.97

Table 2.1: Middlebury Test Bench for discussed algorithms

2.4.5 Analysis

The four discussed algorithms presented in Sections 2.4.1 through 2.4.4, demonstrate various methods of which an accurate disparity map can be constructed. Figures 2.8 through 2.11 give a visual representation of the accuracy achieved by each algorithm. Though there is no mistaking that each discussed algorithm performs extremely well, one other concern that arises is the computation time of each algorithm. Discussing the DoubleBP algorithm, [6] does not specify the computation time or hardware used in their experiments, though it is mentioned that the algorithm was designed to be best suited for parallel hardware acceleration, e.g. the GPU or the IBM's Cell Processor. [6] also states that the computational time of the system depends highly on the total number of iterations taken for the energy function to converge, in the case of the Tsukuba data set, depending on the belief propagation method used, the runtime varies from 3 seconds to 30 seconds when 50 iterations are performed.

The CoopRegion algorithm discussed in [7] achieves slightly less accuracy than that of the DoubleBP algorithm though the computational time of the algorithm is slightly shorter than that of the DoubleBP algorithm. [7] run their algorithm on a notebook computer with a CPU of PM1.6G, but does not clarify whether their algorithm was performed using parallel computing. Similar to that of the DoubleBP, the total computational time of the system is dependant on the total number of iterations done. For the Tsukuba stereo data set, the processing time was approximately 20 seconds, where 4 iterations were done, in addition to the 8 seconds needed

for image segmentation.

The AdaptingBP algorithm presented in [8] is ranked one of the top performing algorithms when the error threshold is set to 1. The AdpatingBP algorithm was run on a 2.21GHz Athlon 64 computer, at which, all four stereo pairs required a time of 14 to 25 seconds of computation time, where the most time consuming process occurred during the mean-shift segmentation step. Again, whether the computational time was computed with parallel computing is not discussed

The ADCensus algorithm presented in [9] achieved the highest accuracy of the four discussed algorithms. ADCensus was tested on a PC with Core2Duo 2.20GHz CPU with a NVIDIA GeForce GTX 480 graphics card. When testing the algorithm over the four stereo pairs in the Middlebury dataset a computation time of 2.5 seconds for Tsukuba, 4.5 seconds for Venus, 15 seconds for Teddy, and 15 seconds for Cones was achieved when CPU implementation was done. Similarly, 0.0016 seconds for Tsukuba, 0.0032 seconds for Venus, 0.0095 seconds for Teddy, and 0.0095 seconds for Cones was needed when GPU implementation was done. For the ADCensus algorithm, the runtime process was mostly consumed by the iterative cost aggregation step and scanline optimization process.

Table 2.1 shows the MiddleBury evaluation table for each algorithm as well as their average error for certain segmnts of each evaluation. Covering the steps taken from each algorithm, majority of the preliminary steps in achieving a confident depth map are similar between each algorithm. Each discussed algorithm employs some method of belief propagation in order to refine the inaccuracies present in preliminary depth maps. Although each discussed algorithm uses a different method of belief propagation, similar results were achieved between each algorithm. Similarly, image segmentation is common among the CoopRegion and AdaptingBP algorithms, where both algorithms implement the mean-shift segmentation algorithm. Similarly, the AdaptingBP and ADCensus algorithms both use a combination of two or more cost

functions to achieve a more accurate result when determining the depth map. By using more than one cost function, the algorithm is capable of achieving higher accuracies by allowing one cost function to progress at portions where typically one cost function would fail. Thus, when choosing two cost functions, the cost functions that are typically chosen tend to compliment one another by having each of them perform better in different scenarios of an image.

2.5 Summary

This chapter discussed several stereo correspondence techniques as well as provided an in depth description of some of the top performing stereo correspondence algorithms. This chapter also provides an analysis of these algorithms, showcasing what differentiates each of them from one another. As seen in Table 2.1, a compilation of each of the algorithms results are made to provide a side by side comparison of all the discussed algorithms.

Chapter 3

Stereo Correspondence using an Assisted Discrete Cosine Transform Method

3.1 Introduction

Stereo matching has become one of the most extensively researched topics in computer vision [6]. For upcoming applications such as Free-viewpoint TV (FTV) and Multiview TV (MTV) stereo matching plays an extremely important role in achieving a high quality experience. There are two major problems that arise with a stereo correspondence algorithm. The first is the matching accuracy of the algorithm, and the second is the computational time. Both FTV and MTV require high accuracy and fast computational time.

This chapter focuses on the adaptation of frequency based features for stereo based matching [10]. The proposed algorithm searches for the top frequency components of each segment, which are then taken and compared across a database as an indicator as to which segment in the database matches the input segment. The top performing algorithms explained in Section 2.4 use color intensity comparisons to determine the preliminary disparity map whereas the

Frequency in Hz	Time in Seconds
823.44	1.054
1892.31	1.321
712.84	1.703
...	...
819.71	9.943

Table 3.1: Example of Fingerprinting a 10 second sample

proposed algorithm uses top magnitude frequency components for the same task. Thus, this section introduces a new initialization to stereo correspondence that deviates from the traditional initial step of common stereo correspondence algorithms to demonstrate the ability of frequency components to accurately find high detailed segments of the image as frequency components provide a more reliable indicator within a target window.

3.2 Motivation

As previously stated, the algorithm presented in [10] determines an audio source by relying on the spectrogram and fingerprinting. A spectrograph is a time-frequency graph and fingerprinting is explained in [10] as identifying the peak intensities and keeping track of the frequency and the amount of time from the beginning of the track that specific frequency occurred at. Table 3.1 provides an example of fingerprinting over a 10 second audio clip. Experiments were done in [10] and conclusions were made that a minimum of 30 points per second were needed for sufficient audio classification, though the number of points can vary.

Figure 3.1 provides an example of a spectrograph and its corresponding constellation map over a 13 second audio clip. As demonstrated, the amount of feature points present within the small audio clip provides plenty of points to use for classification.

As [10] presents a method of constellation maps for the use of audio searching, this method

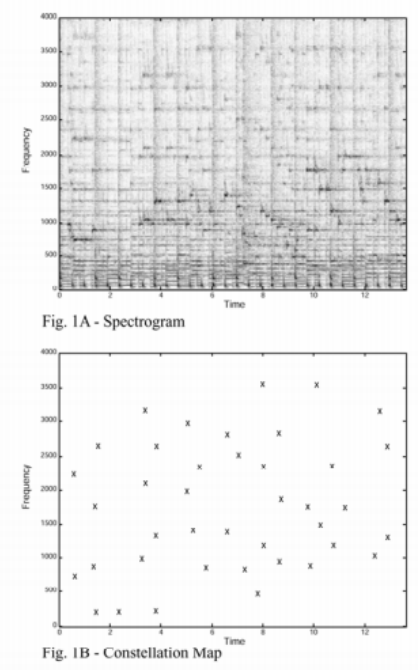

Figure 3.1: Spectrograph example [10]

provided the motivation of using a similar method to that of constellation maps in the image domain. Thus, by taking the concept [10] applied to the audio domain, the proposed algorithm aims to achieve acceptable results in inter-frame stereo correspondence.

3.3 Common Measures

In order to determine corresponding pixels between two images, a form of measuring the similarity or dissimilarity of the target regions must be assessed. Typically, in depth estimation, a dissimilarity measure is used [15]. In this case, a matching cost function, which increases as the similarity between regions decreases. The common notation for matching cost functions are given by C(x,y,d), where (x,y) is the initial position of the target pixel, and d is the disparity between matching stereoscopic images.

Common measures used in pixel matching include the absolute difference (AD), squared intensity difference (SD), and the absolute gradient difference (GRAD), as show in equations 3.1-3.3, respectively.

$$C_{AD}(x,y,d) = |I_L(x,y) - I_R(x-d,y)| \tag{3.1}$$

$$C_{SD}(x,y,d) = |I_L(x,y) - I_R(x-d,y)|^2 \tag{3.2}$$

$$C_{GRAD}(x,y,d) = |\nabla_x I_R(x,y) - \nabla_x I_L(x+d,y| + |\nabla_y I_R(x,y) - \nabla_y I_L(x+d,y| \tag{3.3}$$

Similarly, common measures used in window based matching include the sum of absolute difference (SAD), sum of squared intensity difference (SSD), and the sum of absolute difference SGRAD, shown in equations 3.4-3.6, respectively. The SAD equation takes the sum of the difference between pixels in the original block and the target block. This sum, as seen in Equation 3.4, represents the L^1 norm of the block. Although this is a common method, the main downfall of this cost function is its inability to distinguish lighting variations.

$$C_{SAD}(x,y,d) = \sum_{(u,v)\in W(x,y)} |I_L(u,v) - I_R(u-d,v)| \tag{3.4}$$

The SSD finds the sum of the squared difference values, thus providing emphasis on error allowing it to play a larger role in comparisons than that of the the SAD.

$$C_{SSD}(x,y,d) = \sum_{(u,v)\in W(x,y)} |I_L(u,v) - I_R(u-d,v)|^2 \tag{3.5}$$

Lastly, the SGRAD uses the gradient of the stereoscopic pair to find the similarity between

blocks. Since it uses the gradient, the SGRAD focuses on object edges rather than the textures of the object.

$$C_{SGRAD}(x,y,d) = \sum_{(u,v)\in W(x,y)} |\nabla_x I_R(u,v) - \nabla_x I_L(u+d,v| + \sum_{(u,v)\in W(x,y)} |\nabla_y I_R(u,v) - \nabla_y I_L(u+d,v| \tag{3.6}$$

3.4 Relationship between disparity and depth

Theoretically, the position and orientation of both cameras can be freely chosen, as long as the transformation matrices of both cameras can be found. Then the epipolar lines describing the relationship between both cameras can be easily found through the Principle Matrix as defined in Equation 2.9. In real world applications that incorporate stereo vision, it is necessary to determine the depth of each pixel within a frame as the disparity value only provides an arbitrary value that relates the pixel difference of matching pixels from a pair of stereo images. Thus, this calculation from disparity to depth is particularly trivial assuming accurate disparities are found, as seen in Figure 3.2.

Here, the origins of the left and right camera are defined as O_L and O_R, respectively, and the disparity between corresponding pixels within each frame are defined as $d = x^l - x^r$. Assuming the focal lengths of each camera are equal, it can be seen that the triangle plane is parallel to the ground, and due to the rectification step described in subsection 2.2.1, the vertical position of the projected pixels are the same. Thus, due to this property, the depth of point P from both cameras is only related to the horizontal disparity d, by applying the properties of similar triangles, Equation 3.7 can easily be obtained.

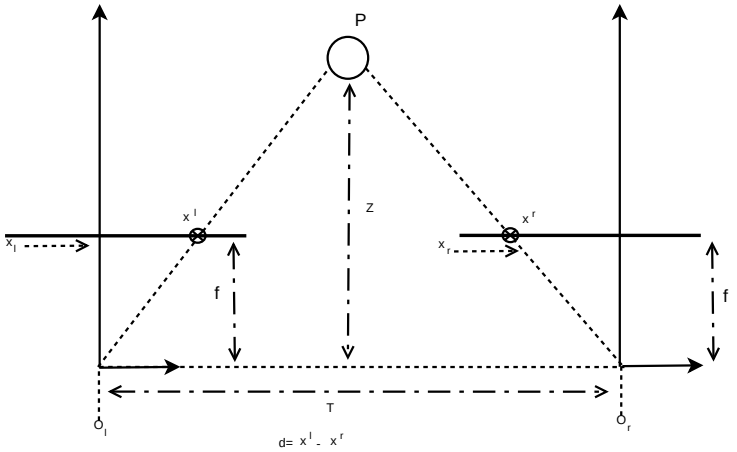

Figure 3.2: Relationship between Disparity and Depth

$$\frac{T-(x^l-x^r)}{T}=\frac{Z-f}{Z} \tag{3.7}$$

Here, by replacing the term $(x^l - x^r)$ with d, and simplifying, Equation 3.7 turns into 3.8.

$$\frac{d}{T}=\frac{f}{Z} \tag{3.8}$$

Thus, by simplifying Equation 3.8 to isolate for Z, it becomes quite trivial that the final equation obtained is that shown in Equation 3.9.

$$Z=f\frac{T}{d} \tag{3.9}$$

As seen in Equation 3.9, the depth of point P, Z, can be found through the relationship between the real world distance, T, the disparity d, and the focal length, f of the left and right cameras.

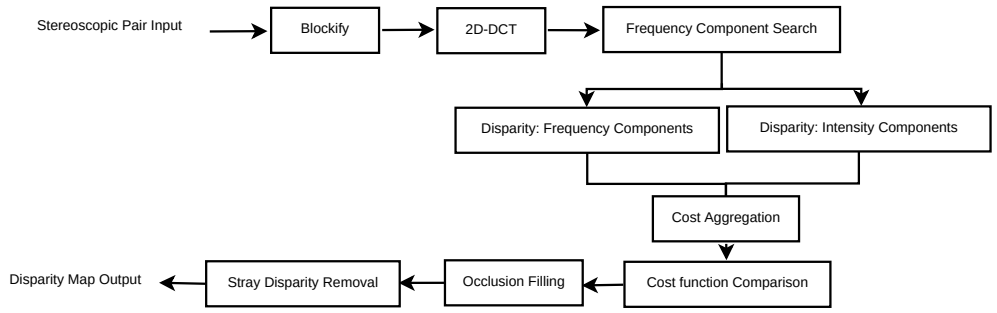

Figure 3.3: Proposed Algorithm Pipeline

3.5 Proposed Method

The proposed algorithm, as shown in Algorithm 1, is designed to follow the similar common pipeline as that of the aforementioned algorithms with a few modifications to improve the performance similar to that of [37, 22, 23], where the full algorithm is schematically presented in Figure 3.3. Conventional algorithms use color intensity of a neighborhood of pixels in determining the most probable disparity, though the accuracy of color intensity comparisons of highly detailed regions of an image can vary. This introduces the deviation of the proposed algorithm from classical algorithms by determining the initial disparity through the frequency component positions and magnitudes of the group of pixels. This allows the disparity of high detailed segments of the image to be found more accurately as frequency components provide a more reliable indicator of the behavior within the target window, this can ultimately be used as a more authentic measure in determining matching pixels. The common intensity comparison is known as a cost function, where the disparity of each target segment is determined by the smallest value among a predetermined search range. These comparisons were discussed in Section 3.3.

Algorithm 1 Proposed Algorithm

DATA: Input Stereo Pair

METHOD: Blockify and DCT
METHOD: Disparity Calculation(I_{left},I_{right})
METHOD: Cost Aggregation

for all blocks **do**
  if $C_{freq} \leq$C$_{mag}$ **then**
    D at block position = D_{freq}
  else if $C_{freq} \geq$C$_{mag}$ **then**
    D at block position = D_{mag}
  else
    $\frac{D_{freq}+D_{mag}}{2}$
  end if
end for
for each pixel in disparity map **do**
  if occluded pixel **then**
    pixel equals previous horizontal pixel
  else
    next pixel
  end if
end for

3.5.1 Discrete Cosine Transform

As explained in [38, 39], the Discrete Cosine Transform (DCT) can be used on an area of an image for feature selection, where the computational time of the DCT is considered to be optimal. The DCT is a special subset of the Discrete Fourier Transform (DFT), where the phase information of the transform is discarded to favor the amplitude information of the image. The DCT feature selection is shown in Equations 3.10,

$$r(x,y,u,v) = \alpha(u)\alpha(v)cos[\frac{(2x+1)u\pi}{2n}]cos[\frac{(2y+1)v\pi}{2n}] \tag{3.10}$$

where u and v are the pixel locations of the target pixel, n is the size of the target window, x and y are the resulting pixel locations, and $\alpha(u)$ and $\alpha(v)$ are the coefficients defined in 3.11.

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{n}} & \text{for } u\text{=0} \\ \sqrt{\frac{2}{n}} & \text{for u=1,2...n-1} \end{cases} \tag{3.11}$$

3.5.2 Pixel Matching

The proposed method represents the combined array shown in Equations 3.12 through 3.14 where N is the window size, λ is the weight that determines the amount of influence the position of the maximum frequency components have on the comparison of target arrays. With the concatenated array of frequency positions and magnitudes, an accurate disparity map for highly detailed regions can be found. To ensure that frequency components are only effected by the pixels within the window, the two dimensional DCT is taken within the target pixel window. Thus, I_{freqL} is the two dimensional DCT window of the target pixel. Similarly, Equations 3.12 through 3.14 are also performed on the target two dimensional DCT window in the right image.

$$position_L(u,v) = pos(max_N(I_{freqL})) \tag{3.12}$$

$$magnitude_L(u,v) = mag(max_N(I_{freqL})) \tag{3.13}$$

$$arr_L(u,v) = concat(\lambda * position_L(u,v), (1-\lambda) * magnitude_L(u,v)) \tag{3.14}$$

Once each target array is determined, Equations 3.15 and 3.16 are calculated over each window to determine the initial disparity map derived from frequency components, where C_{freq} is the frequency cost function and D_{freq} is the corresponding disparity map.

$$C_{freq}(u,v,d) = |arr_L(u,v) - arr_R(u-d,v)| \tag{3.15}$$

$$D_{freq}(u,v) = min(C_freq(u,v,:)) \tag{3.16}$$

The resulting disparity map is still very inconsistent in regions of low details. To accommodate for these low detailed regions, the intensity value SAD is implemented. In order to ensure the stability of each cost function, the cost aggregation algorithm proposed in [9], which was described in Section 2.4.4, is adopted. The three enhanced rules in [9] are placed when performing cost aggregation to ensure that the color between pixels are similar, and to allow more flexibility in the production of the arm lengths. This cost aggregation algorithm is iterated four times to ensure that stable cost values are obtained, although it is possible to reduce the amount of iterations to achieve faster computational times.

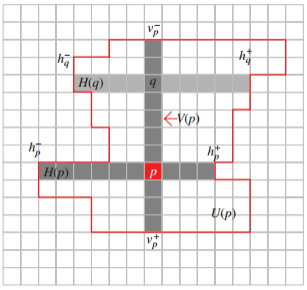

Figure 3.4: Cost Aggregation

3.5.3 Cost Aggregation

Cost Aggregation as proposed in [35] looks to find an appropriate local support region for each pixel. This local support region contains neighboring pixels from the same disparity as the target pixel. The assumption behind this is that pixels with similar intensity values within a local area are commonly from the same structure, therefore having similar disparity values. This upright cross, a search area defined by the same disparity values in the horizontal and vertical directions, have a big potential to reduce computation redundancy. An example of an upright cross for a target pixel, p, is shown in Figure 3.4. Here, the upright cross for the target pixel, p, shown by the dark gray arrays, where the stopping point of the cross is determined by differentiating pixel values.

3.5.4 Cost Normalization

In order to ensure that the intensity cost function does not interfere with the calculated frequency regions, each cost function is normalized and compared. Here 3.17 normalizes the magnitude cost function, where C_{Mag} is the intensity cost function found through the magnitude comparisons performed through one of the Equations in Section 3.3.

$$C_{mag} = C_{Mag}/max(C_{Mag}) \tag{3.17}$$

3.18 normalizes the DCT cost function, where C_{freq} is the frequency intensity cost function.

$$C_{freq} = C_{freq}/max(C_{freq}) \tag{3.18}$$

Once each cost function is normalized, D, the resulting proposed disparity map is found through the cases defined in 3.19. Here, D_{Mag} is the corresponding magnitude disparity map, D_{avg} is the average of D_{Mag} and D_{freq} at the target pixel, and τ is the predetermined difference threshold set by the user. Thus, by normalizing each cost function to 1, the cases shown in Equation 3.19 provide the optimal disparity decision for the target pixel.

$$D(u,v) = \begin{cases} D_{freq}(u,v) & C_{freq} < C_{Mag}(u,v) \\ D_{Mag}(u,v) & C_{Mag} < C_{freq}(u,v) \\ D_{avg}(u,v) & |C_{Mag} - C_{freq}(u,v)| < \tau \end{cases} \tag{3.19}$$

3.5.5 Occlusion Filling

As described in Section 1.4.1, occlusions can become a big problem when trying to achieve accurate depth maps. In order to alleviate this problem, [40] presents several occlusion filling methods. Here, occlusion filling is the term used as most, if not all, algorithms handle occlusions through the projection of known pixels onto the occluded regions. [40] describes four different occlusion filling methods ranging from simple neighborhood comparisons to complex probability statistics.

Occlusion filling is carried out to remove any occlusions that are present between the two pairs of stereo images [41]. The process uses the determined disparity map from the previous

step and shift each pixel in the left image by the corresponding pixel defined by the disparity map, as shown in equation 3.20, where Im_{right} is the virtually generated right image, and I_{left} is the original stereo image.

$$Im_{right}(u,v) = I_{left}(u+d,v) \tag{3.20}$$

Once the virtual right image is generated, it is subtracted from the original right image, and an occlusion map is formed based on the difference matrix as defined in equation 3.21 and 3.22, where diff is the difference matrix determined by the subtraction of the virtual image from the real image, occ is the defined occlusion map, and τ_{occ} is the pixel difference needed to define an occlusion.

$$diff = |I_{right} - Im_{right}| \tag{3.21}$$

$$occ(u,v) = \begin{cases} 1 & diff(u,v) > \tau_{occ} \\ 0 & \text{otherwise} \end{cases} \tag{3.22}$$

After the occlusion map is found, the Neighbor's Disparity Assignment (NDA) occlusion method defined in [42] is used. As described in section 1.4.1, the two occlusion types, border occlusions and non-border occlusions are handled by the NDA occlusion method explained in [42].

3.5.6 Noise Removal

The last step in the proposed algorithm is to remove any stray disparities that were neglected through the previous steps of the algorithm. This removal process is done using a one dimen-

sional filter that compares the target pixel with its two horizontal neighbors. The comparison is performed based on three cases, as shown in equations 3.23 and 3.24.

$$D_x(u,v) = \frac{D(u-1,v)+D(u+1,v)}{2} \tag{3.23}$$

$$D(u,v) = \begin{cases} D(u,v) & D(u,v) \neq D(u-1,v), D(u,v) = D(u-1,v) \\ D(u,v) & D(u,v) = D(u-1,v), D(u,v) \neq D(u-1,v) \\ D_x(u,v) & D(u,v) \neq D(u-1,v), D(u,v) \neq D(u-1,v) \end{cases} \tag{3.24}$$

The first case shown in equation 3.24 is that the disparity of the target pixel is the same as that of the pixel to its right. In this case the disparity value is left alone. Similarly, if the disparity value of the target pixel is the same as that of the pixel to its left, it is also left alone. The third case in which the target pixel disparity is not the same as either of its neighbors, the average disparity of the two horizontal neighbors is taken. Here average is preferred to taking the median of the filter because the stray disparity value of the target pixel becomes very unpredictable, thus taking the median of the filter can result in the disparity value not changing whereas taking the average of the two neighbors will result in a more accurate disparity value assuming that the neighboring pixels are correct.

3.6 Summary

This chapter introduced the proposed method that uses frequency components in determining the disparity map. As described over the chapter, the proposed algorithm uses an assisted Discrete Cosine Transform in determining the disparity map of a stereo pair. This chapter also explained the other necessary steps to achieve the proposed algorithm as seen in Figure 3.3.

Chapter 4

Experimentation and Discussion

4.1 Steresocopic images

There are plenty of images that can be used for disparity matching, though the most common stereoscopic pairs used for algorithm analysis are the 2001, 2003, 2005, and 2006 datasets found in the Middlebury database [1]. Figure 4.1 displays several stereoscopic pairs found in [1] that are typically used. Here, these images are normalized to have matching intensities between each stereoscopic pair, removing any noise and biasing that may occur from individual cameras. Secondly, all these images are rectified to remove any vertical ambiguity between the pairs for reasons previously mentioned in section 2.2.1.

These images were chosen due to their popularity of testing among the stereo correspondence community surveyed in [1], the ease of accessibility, and the lack of need to perform any preprocessing, rectification, and camera calibration. Additionally, these four images are regarded as the golden standard benchmark for stereo correspondence algorithms, thus making the availability of comparisons and maximum disparities much more accessible then those of other stereo pairs.

Figure 4.1: a) b) Tsukuba stereoscopic pair, c) d) Venus stereoscopic pair, e) f) Teddy stereoscopic pair, g) h) Cones stereoscopic pair

4.2 Ground Truth Evaluation

The error evaluation of the proposed algorithm was determined using the ground truth depth maps and the maximum disparity levels for each stereoscopic pair provided by the Middlebury Stereo homepage [1]. The ground truth disparity maps of some of the images are shown in Figure 4.2. Here, the disparity errors are found by scaling the ground truth image to the maximum disparity of the stereo pair and subtracting it from the disparity map found from the proposed method, as shown in Equation 4.1.

$$D_c = MD * \frac{D_t}{255} - D_p \tag{4.1}$$

Where D_c is the resulting difference disparity map, MD is the Maximum disparity of the stereoscopic pair, D_t is the ground truth disparity map, and D_p is the proposed disparity map. Once the ground truth image is subtracted from the proposed disparity map, any absolute difference greater than one is regarded as an error whereas any absolute difference smaller than one is regarded as correct, these cases can be seen in Equation 4.2, where D_{diff} is the difference error disparity map and α_{occ}. is the disparity marginal error.

$$D_{diff}(x,y) = \begin{cases} 0 & D_c(x,y) > \alpha_{occ} \\ 255 & D_c(x,y) < \alpha_{occ} \end{cases} \tag{4.2}$$

This disparity marginal error is flexible, though the evaluation of algorithms in [1] also changes. Therefore, increasing the acceptable threshold for correct pixels not only reduces the top performing pixel error rate but it may also change the top performing algorithm. For the

evaluation carried out in this thesis, the threshold pixel difference for acceptable disparities is locked to one.

4.3 Results and Discussion

In order to verify the effectiveness of the proposed algorithm, it was implemented in Mathworks MATLAB and several variable values were tested to ensure the optimal results were achieved. Variable values of τ and τ_{occ} were experimentally determined as 0.2 and 25, respectively to give optimal performance. The algorithm was tested on the four different stereoscopic pairs in Figure 4.1 which are provided by the Middlebury database [1]. Table 4.1 summarizes the variables for each set of stereoscopic pairs. In the table, τ is the predetermined difference threshold determined by the user when each cost function is normalized to 1, τ_{occ} is the pixel difference needed to define an occlusion, the weight column defines the weight of the position of the maximum frequency against the weight of the magnitude of the maximum frequency point, and maximum disparity and scale are the maximum accurate disparity available for each stereo pair and the appropriate scaling factor, respectively. The four stereoscopic pairs are the Teddy, Venus, Tsukuba, and Sawtooth pairs, where each stereoscopic pair achieved an average pixel error rate of 7.1%, 4.5%, 5.2%, and 2.6% respectively. Figure 4.3 shows the disparity map obtained from the proposed algorithm for the Venus stereoscopic pair alongside the Venus disparity results for the discussed algorithms in section 2.4, where the slight increase in error rate compared to these algorithms are caused by the miniscule changes between disparities. Figure 4.4 demonstrates the resulting disparity maps for the Cones and Teddy stereo pairs beside each of their respective ground truth disparity maps. Figure 4.5 shows the incorrect disparity values in the Teddy and Cones datasets when compared with the ground truth disparity maps. As shown in black, the areas that cause the algorithm to misclassify the target pixel are caused

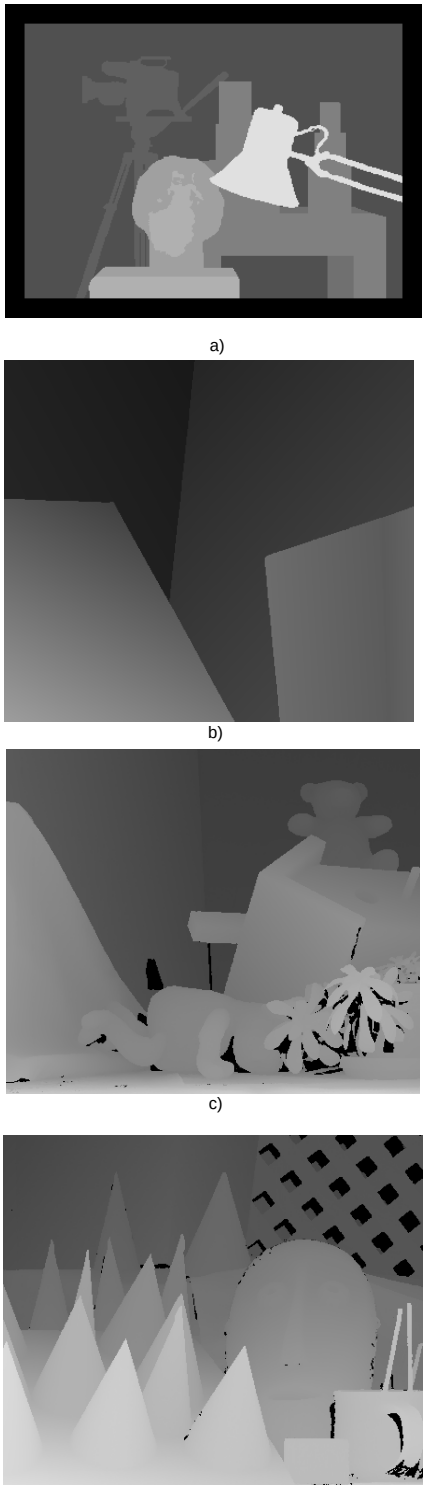

Figure 4.2: Ground Truths of the left image for a) Tsukuba, b) Venus, c) Teddy , d) Cones

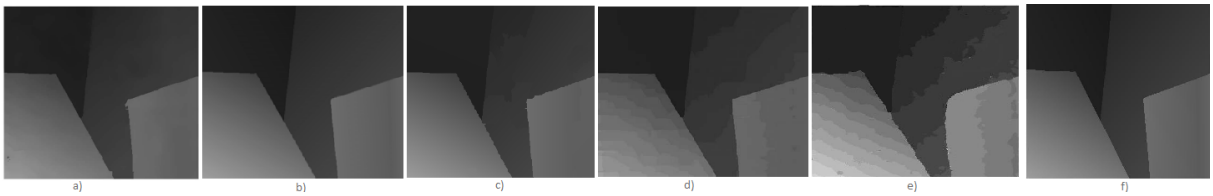

Figure 4.3: Comparison of the Venus Stereoscopic pair: a) Result from DoubleBP. b) Result from CoopRegion. c) Result from AdaptingBP. d) Result from ADCensus. e) Result from Proposed Algorithm f)Ground Truth Disparity Map

Stereo Pair	τ	τ_{occ}	Weight (τ)	Maximum Disparity	Scale
Tsukuba	0.2	25	0.75	16	16
Venus	0.2	25	0.9	20	8
Teddy	0.2	25	0.9	60	4
Cones	0.2	25	0.6	60	4

Table 4.1: Variable Definitions and Values

mainly by the regions that converge from one disparity to the next due to the use of integer disparities in the proposed algorithm, though it is believed that these misclassifications can be corrected if more robust methods for occlusion filling and noise filtering are used.

Table 4.2 compares the average pixel error rate of the proposed algorithm with the top performing algorithms in the Middlebury evaluation [1]. Experimentation was done using a window size of 9x9, thus a maximum of 9 frequency points were taken for frequency com-

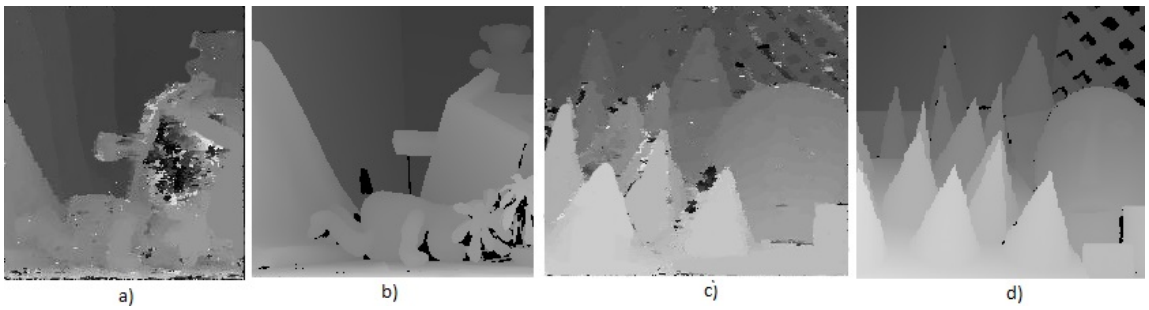

Figure 4.4: Comparing the results of the proposed algorithm on the Cones and Teddy stereo pairs with the ground truths a), c)Results of proposed algorithm. b), d)Ground truth disparity maps

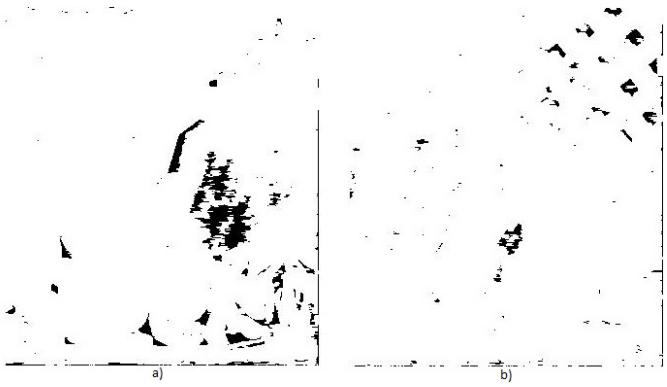

Figure 4.5: Incorrect disparity values of the a)Teddy stereo pair and b)Cones stereo pair marked as a black pixel

parisons, whereas compared to the algorithm proposed by [7], the algorithm takes up to 30 frequency components per second. Table 4.2 also compares the average computational time of the proposed algorithm with the classical stereoscopic algorithms. As seen, the computational time of the proposed algorithm is significantly lower than that of the traditional stereo correspondence algorithms. This is mainly due to the elimination of the computationally expensive algorithms, such as mean-shift segmentation and belief propagation, present in the other algorithms. The computational time can be further reduced through careful GPU implementation of the proposed method.

Computation times in brackets in Table 4.2 are the normalized computational times of the proposed and traditional stereo correspondence methods. This was done using the benchmarking method in Matlab alongside the available system specifications provided in [6], [7], [8], and [9]. Normalization was done by normalizing all other algorithms to the algorithm with the lowest system specifications. As seen, even after normalization, the proposed method still outperforms the computation time of the traditional methods.

Algorithm	Teddy	Venus	Tsukuba	Cones	Avg. pixel error rate (%)	Avg. Computation Time (Normalized)
DoubleBP	8.30%	0.45%	1.28%	8.78%	4.19%	15 sec (15 sec)
CoopRegion	8.31%	0.21%	1.16%	7.18%	4.41%	20 sec (20 sec)
AdaptingBP	7.06%	0.21%	1.37%	7.92%	4.23%	18 sec (19 sec)
ADCensus	6.22%	0.25%	1.37%	7.25%	3.97%	10 sec (11 sec)
Proposed Algorithm	**7.1%**	**0.3%**	**1.5%**	**7.4%**	**4.07%**	**5 sec (8 sec)**

Table 4.2: Comparison of pixel error rates and computation times

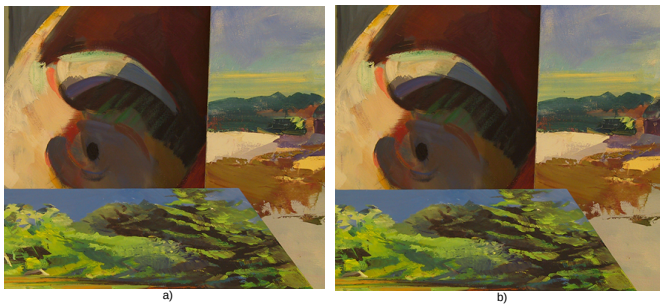

Figure 4.6: Bull Stereo Pair

4.4 Additional Test Images

In addition to the four golden standard images used in typical stereo correspondence algorithms, the proposed algorithm was also tested on additional images to verify the applicability of the algorithm to a wider range of stereo image pairs. The Bull stereoscopic pair is shown in Figure 4.6, and the comparison of the proposed disparity map alongside the ground truth disparity map are illustrated in Figure 4.7. It should be noted that the process of calculating an accurate disparity map for images outside of the four golden standard images are quite difficult due to the lack of available information. In the case of the Bull stereo pair, a maximum disparity value of 20 was used, with τ and τ_{occ} being the same as those from the golden standard images, and a weight value of 0.9. As illustrated in Figure 4.7, the majority of the errors come from the transition of one disparity to another.

The second example is the sawtooth stereoscopic pair shown in Figure 4.8, which uses a maximum disparity value of 12, with τ and τ_{occ} being the same as those from the golden stan-

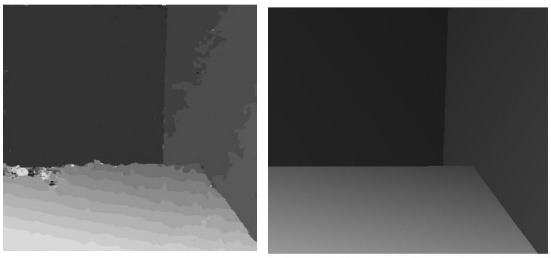

Figure 4.7: Bull Comparison

Figure 4.8: Sawtooth Stereo Pair

dard images. The maximum disparity values for both the bull stereo pair and the sawtooth stereo pair are not given in the stereo correspondence database in [1], thus they must be manually calculated based on the assumptions that were discussed in Section 1.4.3. The disparity map of the sawtooth stereoscopic pair obtained from the proposed method is given in Figure 4.9.

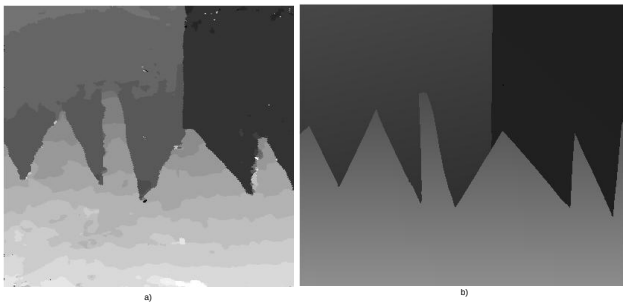

Figure 4.9: Sawtooth Comparison

Stereo Pair	τ	τ_{occ}	Weight (τ)	Maximum Disparity	Scale
Bull	0.2	25	0.75	12	22
Sawtooth	0.2	25	0.9	12	22

Table 4.3: Variable Definitions for additional images

Table 4.3 shows the variable definitions for the extra images that the proposed method was tested on. The maximum disparity, and scale columns are both experimentally found based on the 3% restriction that was discussed in Section1.4.3. The scale value was determined by finding the closest integer value that would scale the disparity map from its maximum disparity to the usual 255 that is seen in images.

4.5 Summary

This chapter first discussed the stereo image pairs used for testing. It then presented experiments that were conducted using the proposed algorithm and compared them with the results of traditional methods explained in Chapter 2. As shown in Table 4.2, the proposed algorithm performs well when compared with the other algorithms. The differentiating factor of the proposed algorithm, in cotnrast to classical stereo correspondence algorithms is the computational time for a disparity map to be made, where the proposed method achieves a disparity map in at least half the time of the next competing algorithm.

In addition to the 4 golden standard images used for testing in stereo correspondence algorithms, the algorithm was also tested on a set of additional images to demonstrate its feasibility for a wider range of videos. The resulting disparity maps for these additional images were shown in Figures 4.7 and 4.9 with their variable declarations summarized in Table 4.3.

Chapter 5

Conclusions

This thesis studied into detail explaining key aspects that must be considered to achieve a successful disparity map. Then it presented the top performing algorithms found [1] and discusses their advantages and shortcomings. The central part of the thesis is the proposal and implementation of a stereo correspondence algorithm which is capable of providing results that are comparable to the state-of-the-art algorithms found in the Middlebury database [1] for the typical four images used for stereo correspondence testing which significantly reduced computation times. These four images being the Teddy, Cones, Tsukuba, and Venus stereo pairs.

As shown in Chapter 4, the use of a block based DCT parallel to the traditional color intensity cost function demonstrates that the quality of results are on par with the top performing algorithms while achieving a computational time that is at least half of that of the next fastest algorithm. The proposed algorithm uses aspects from the DoubleBP [6], CoopRegion [7], AdaptingBP [8], and ADCensus [9] algorithms. To expand on this more, the proposed algorithm takes contributing factors that distinguish each of the state-of-the-art algorithms and uses them to improve the overall performance.

Future work to be done is to implement an occlusion filling method that is more robust and is capable of determining occlusions based off the probability of surrounding pixels [42]. Currently, an NDA occlusion filling method is used but [40] provides a set of other occlusion methods that may provide better disparity results. Such occlusion methods include the *Weighted Least Squares* (WLS) method, *Diffusion in Intensity Space* (DIS), and *Segmentation-based Least Squares* (SLS) occlusion methods. Additionally, a method to achieve more frequency points within a given window which would potentially lead to a more discriminative selection of matching pixels. This study would align more with the motivating work proposed in [10].

Publications

Conferences and Workshops

1. **Edward Rosales**, and L. Guan. Stereo Correspondence Using an Assisted Discrete Cosine Transform. *IEEE Visual Communications and Image Processing (IEEE-VCIP 2014)*. [Accepted]

2. Y. He, Z. Zhang, X. Nan, N. Zhang, F. Guo, **Edward Rosales** and L. Guan. vConnect: Connect the Real World to the virtual World. *IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA 2014)*. pp. 30-35

3. **Edward Rosales**, Y. Tie, A. Venetsanopoulos and L. Guan. Automatic Face Recognition from Video Sequences using a Template Based Cross Correlation Method. *Canadian Conference of Electrical and Computer Engineering 2013 (CCECE 2013)*. pp. 1-4

References

[1] R. S. D. Scharstein. http://vision.middlebury.edu/stereo/. website. [Online]. Available: http://vision.middlebury.edu/stereo/

[2] B. Wilburn, N. Joshi, V. Vaish, E. ville Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Trans. Graph*, pp. 765–776, 2005.

[3] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *ACM SIGGRAPH 2004 Papers*, ser. SIGGRAPH '04. New York, NY, USA: ACM, 2004, pp. 600–608. [Online]. Available: http://doi.acm.org/10.1145/1186562.1015766

[4] T. Kanade, P. Rander, and P. Narayanan, "Virtualized reality: constructing virtual worlds from real scenes," *MultiMedia, IEEE*, vol. 4, no. 1, pp. 34–47, Jan 1997.

[5] M. Tanimoto, M. Tehrani, T. Fujii, and T. Yendo, "Free-viewpoint tv," *Signal Processing Magazine, IEEE*, vol. 28, no. 1, pp. 67–76, Jan 2011.

[6] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *Pattern*

Analysis and Machine Intelligence, IEEE Transactions on, vol. 31, no. 3, pp. 492–504, 2009.

[7] Z.-F. Wang and Z.-G. Zheng, "A region based stereo matching algorithm using cooperative optimization," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1–8.

[8] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3, 2006, pp. 15–18.

[9] X. Mei, X. Sun, M. Zhou, shaohui Jiao, H. Wang, and X. Zhang, "On building an accurate stereo matching system on graphics hardware," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 467–474.

[10] A. L. Wang, "An industrial-strength audio search algorithm," in *ISMIR 2003, 4th Symposium Conference on Music Information Retrieval*, 2003, pp. 7–13, in , S. Choudhury and S. Manus, Eds., The International Society for Music Information Retrieval. http://www.ismir.net: ISMIR, October , pp. . [Online]. Available: http://www.ee.columbia.edu/ dpwe/papers/Wang03-shazam.pdf.

[11] D. M. Regan, *Human Perception of Objects: Early Visual Processing of Spatial Form Defined by Luminance, Color, Texture, Motion and Binocular Disparity:*, 1st ed. Sinauer Associates Inc., March 2000.

[12] G. Yao, Y. Liu, B. Lei, and D. Ren, "A rapid stereo matching algorithm based on disparity interpolation," in *World Automation Congress (WAC), 2012*, 2012, pp. 5–10.

REFERENCES

[13] X. Yu and X. Rong, "Pseudo disparity based stereo image coding," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, Oct 2008, pp. 2444–2447.

[14] J. Konrad and Z.-D. Lan, "Dense-disparity estimation from feature correspondences," pp. 90–101, 2000. [Online]. Available: http://dx.doi.org/10.1117/12.384433

[15] A. Olofsson, "Modern stereo correspondence algorithms: Investigation and evaluation," Master's thesis, Linkoping University, 2010.

[16] M. Tanimoto, "Overview of ftv (free-viewpoint television)," in *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, June 2009, pp. 1552–1553.

[17] M. Tanimoto and M. Wildeboer, "Frameworks for ftv coding," in *Picture Coding Symposium, 2009. PCS 2009*, May 2009, pp. 1–4.

[18] M. Tanimoto, "Ftv (free-viewpoint television) for ray and sound reproducing in 3d space," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, pp. 5441–5444.

[19] T. Naemura and H. Harashima, "Ray-based approach to integrated 3d visual communication," in *Goddard Space Flight Center (http://uav.wff.nasa.gov*. Springer Verlag, 2000, pp. 183–213.

[20] M. Schmeing and X. Jiang, "Depth image based rendering," in *Pattern Recognition, Machine Intelligence and Biometrics*, P. Wang, Ed. Springer Berlin Heidelberg, 2011, pp. 279–310. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-22407-2_12

[21] C. Jung and L. Jiao, "Reliable depth-image-based rendering using parameter approximation in mobile devices," *IEICE Electronics Express*, vol. 7, no. 10, pp. 666–671, 2010.

[22] X. Liu, Z. Lei, Q. Yu, X. Zhang, Y. Shang, and W. Hou, "Multi-modal image matching based on local frequency information," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, no. 1, p. 3, 2013. [Online]. Available: http://asp.eurasipjournals.com/content/2013/1/3

[23] P. Vandewalle, S. Süsstrunk, and M. Vetterli, "A frequency domain approach to registration of aliased images with application to super-resolution," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–14, March 2006. [Online]. Available: http://rr.epfl.ch/3/

[24] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, Nov 2000.

[25] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007, pp. 1–8.

[26] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proceedings of the third European conference on Computer Vision (Vol. II)*, ser. ECCV '94. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1994, pp. 151–158. [Online]. Available: http://dl.acm.org/citation.cfm?id=200241.200258

[27] A. Z. R. Hartley, *Multiple View Geometry in Computer Vision*, 2nd Edition, Ed. Cambridge University Press, 2002.

[28] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2010.

[29] M. Tanimoto, "Ftv (free-viewpoint tv)," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, Sept 2010, pp. 2393–2396.

REFERENCES

[30] C. L. Zitnick and S. B. Kang, "Stereo for image-based rendering using image over-segmentation." *International Journal of Computer Vision*, vol. 75, no. 1, pp. 49–65, 2007. [Online]. Available: http://dblp.uni-trier.de/db/journals/ijcv/ijcv75.html#ZitnickK07

[31] A. Isaksen, L. McMillan, and S. J. Gortler, "Dynamically reparameterized light fields," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 297–306. [Online]. Available: http://dx.doi.org/10.1145/344779.344929

[32] M. Gerrits and P. Bekaert, "Local stereo matching with segmentation-based outlier rejection," in *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*, 2006, pp. 66–66.

[33] M. A. F. R. C. Bolles, "Random sample conses: A paradigm for model fitting with applications to image analysis and automated cartography," in *Coomun.. ACM, vol. 24,no. 6*, 1981.

[34] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

[35] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 7, pp. 1073–1079, 2009.

[36] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 2, pp. 328–341, 2008.

[37] B. Reddy and B. N. Chatterji, "An fft-based technique for translation, rotation, and scale-invariant image registration," *Image Processing, IEEE Transactions on*, vol. 5, no. 8, pp. 1266–1271, Aug 1996.

[38] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[39] N. Ahmed, T. Natarajan, and K. Rao, "Discrete cosine transform," *Computers, IEEE Transactions on*, vol. C-23, no. 1, pp. 90–93, Jan 1974.

[40] S. Huq, A. Koschan, and M. A. Abidi, "Occlusion filling in stereo: Theory and experiments." *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 688–704, 2013.

[41] C. Zitnick and T. Kanade, "A cooperative algorithm for stereo matching and occlusion detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 7, pp. 675–684, 2000.

[42] S. Cho, I. Sun, J. Ha, and H. Jeong, "Occlusion detection and filling in disparity map for multiple view synthesis," in *Computing and Networking Technology (ICCNT), 2012 8th International Conference on*, 2012, pp. 425–432.