Fall 12-2010

# Stereo Matching Using a Modified Efficient Belief Propagation in a Level Set Framework

Stephen Goyer Rogers
*University of Southern Mississippi*

The University of Southern Mississippi

# STEREO MATCHING
# USING A MODIFIED EFFICIENT BELIEF PROPAGATION
# IN A LEVEL SET FRAMEWORK

by

Stephen Goyer Rogers

Abstract of a Dissertation
Submitted to the Graduate School
of The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

December 2010

ABSTRACT

STEREO MATCHING
USING A MODIFIED EFFICIENT BELIEF PROPAGATION
IN A LEVEL SET FRAMEWORK

by Stephen Goyer Rogers
December 2010

Stereo matching determines correspondence between pixels in two or more images of the same scene taken from different angles; this can be handled either locally or globally. The two most common global approaches are belief propagation (BP) and graph cuts.

Efficient belief propagation (EBP), which is the most widely used BP approach, uses a multi-scale message passing strategy, an O(k) smoothness cost algorithm, and a bipartite message passing strategy to speed up the convergence of the standard BP approach. As in standard belief propagation, every pixel sends messages to and receives messages from its four neighboring pixels in EBP. Each outgoing message is the sum of the data cost, incoming messages from all the neighbors except the intended receiver, and the smoothness cost. Upon convergence, the location of the minimum of the final belief vector is defined as the current pixel's disparity.

The present effort makes three main contributions: (a) it incorporates level set concepts, (b) it develops a modified data cost to encourage matching of intervals, (c) it adjusts the location of the minimum of outgoing messages for select pixels that is consistent with the level set method.

When comparing the results of the current work with that of standard EBP [11], the disparity results are very similar, as they should be.

The University of Southern Mississippi

# STEREO MATCHING
# USING A MODIFIED EFFICIENT BELIEF PROPAGATION
# IN A LEVEL SET FRAMEWORK

by

Stephen Goyer Rogers

A Dissertation
Submitted to the Graduate School
of The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Approved:

Dr. Beddhu Murali
Director

Dr. Cliff Burgess

Dr. Ray Seyfarth

Dr. Andrew Strelzoff

Dr. Joe Zhang

Dr. Susan Siltanen
Dean of the Graduate School

December 2010

## ACKNOWLEDGMENTS

I want to take this opportunity to thank my advisor, Dr. Beddhu Murali, my committee members, Dr. Ray Seyfarth, Dr. Cliff Burgess, Dr. Joe Zhang, and Dr. Andrew Strelzoff, and everyone else who helped me get to this point.

# TABLE OF CONTENTS

# LIST OF ILLUSTRATIONS

**Figure**

# LIST OF TABLES

**Table**

# LIST OF ABBREVIATIONS

**EBP** - Efficient Belief Propagation [11]
**mEBP** - modified Efficient Belief Propagation
**AD** - Absolute Difference
**BT** - Birchfield-Tomasi's pixel Dissimilarity [2]
**MM** - Level Set min/max operations
**IDC** - Initial Data Cost
**HBP** - Heirarchial Belief Propagation
**PoT** - Percentage of Total
**PoPts** - Percentage of Points

# Chapter 1
# INTRODUCTION

## 1.1 What is Stereo Matching?

Stereo matching is the process of determining correspondence between pixels within two or more images of a single scene taken from different angles. However, using image rectification, whereby images are typically horizontally aligned, the general stereo vision problem can be simplified to finding only the horizontal correspondence, especially for narrow baseline stereo. For non-stationary scenes, multiple views are typically used to produce optimal stereo matching results. However, for stationary scenes, such as those used in this work, multi-view does not offer a significant advantage over two-view. In addition, the two-view approach is computationally less expensive and requires less memory.

In two-view stereo, the two images are typically called the left and the right images. One can either find the correspondence of the left image pixels, called left correspondence, in the right image or the correspondence of the right image pixels, called the right correspondence, in the left image. When finding the left correspondence, the left image is called the reference image and the right image is called the search image and vice versa. The amount of shift between corresponding pixels is called disparity, usually expressed in terms of the number of pixels between corresponding points. Thus, one can compute either the right disparity field (with the right image as the reference) or the left disparity field (with the left image as the reference). While the disparity value can be treated as a real number, assuming it to be an integer simplifies calculations and storage.

## 1.2 Review of Previous Work

Numerous approaches have been developed to handle stereo vision; both locally and globally based on the search area.

### 1.2.1 Local-Based Approaches

In the case of local approaches, the search area for finding a match is confined to a fixed-size window within the search image; for additional discussion, see Murali [29]. While the local-based approaches may be faster, the results are often less impressive than those of the global approaches due to the presence of "streaks" especially in older local methods. Most of the recent local approaches have sought to combine the accuracy found in the global approaches with the efficiency of the local approaches.

In Yoon and Kweon [51], the cost function is calculated using a weighted window-based approach, where each pixel in the window is assigned a specific weight using both the color and proximity differences from the center of the window. First, weights are computed for both the left and right windows. Then the window score is computed according to $WS(p,p\prime) = \frac{\sum_{q\in N_p q\prime\in N_{p\prime}} w^L(p\prime,q\prime) * w^R(p,q) * |I^R_q - I^L_{q\prime}|}{\sum_{q\in N_p q\prime\in N_{p\prime}} w^L(p\prime,q\prime) * w^R(p,q)}$, where p and q represent the current pixel and one of its neighboring pixels in the reference image window, while $p\prime$ and $q\prime$ represent the corresponding points in the search image for p and q. The advantage of Yoon and Kweon's approach is that it produces very good disparity results in spite of being a local method. On the other hand, the major disadvantage is that computationally it takes much longer to compute the disparity field than the global methods. Thus, this method is of little practical use in applications that demand real-time performance. Nister et al. [49] adopted the approach of Yoon and Kweon in their global approach to calculate an initial disparity field. However, they replaced the absolute difference with the Birchfield-Tomasi dissimilarity measure [2]. It has been shown in the literature that BT is more robust to image sampling errors and produces better results than absolute difference. Additionally, [2] also performs its operations with only a 10 percent increase in time over absolute difference.

Another idea that is widely used in stereo vision literature is the truncated absolute difference(TAD), $min(|I^R_{x,y} - I^L_{x-d,y}|, \tau)$, where $\tau$ is a user specified parameter. Recently, for example, Tomabari, Mattoccia, and Di Stefano [42] and Tombari et al. [43] have used

TAD along with an image segmentation to speedup the weight calculation in their local algorithm.

In Mattoccia, Giardino, and Gambini [26], the goal is to combine the accuracy found in those approaches using "adaptive weights" with the efficiency of more traditional "correlative" approaches. To this end, this approach divided each window into blocks, utilizing Yoon and Kweon's method [51] to calculate spatial filtering within each block and calculating a color average for the range filtering. The block based weights, along with the pixel-based matching cost, are combined to yield an overall score with fewer individual spatial and range filtering values. While improving robustness to noise, this approach's pixel-wise cost computation and block averaging shows a speedup through the use of integral image [8, 45] and box filtering schemes. According to [26], the results are ranked highly amongst the top performers for both segmentation and adaptive weights, while requiring less computation.

Mattoccia [21] utilizes "mutual relationships between neighboring pixels," while "explicitly modeling continuity constraints." The accuracy of this approach is very high especially in areas of depth discontinuity and low texture. There are two versions of this approach; combining [21] with fast bilateral stereo and combining [21] with a fixed window. Both approaches performed extremely well, the first outperforming all others in its area, and the other approach ranking second. [21] also states that the second version provides a "trade-off between accuracy and efficiency."

Unlike traditional scanline-based local approaches, Hirschmuller [14] overcomes the issue of streaking through the incorporation of global information in its energy function calculations to determine correct correspondence. However, while it provides accuracy at depth boundaries as well as "uniform regions," this approach is unable to enforce the ordering constraint between neighboring pixels. Hirschmuller [14] also states that its high memory requirement is another drawback.

Mattoccia, Tombari, and Di Stefano [27] "combine an effective cost aggregation strat-

egy with a scanline optimization based disparity computation framework." In this method, the cost is calculated using a segmentation support. The disparity calculation depends on the scanline optimization process, collecting information from four directions (top, bottom, left, and right). Despite its slow runtime, due to the cost aggregation, this method produces good results.

In Mattoccia [24], the author utilizes Mattoccia [21] to improve the accuracy of the initial disparity for traditional scanline optimization/dynamic programming algorithm. In this approach, [21] is used to create two independent disparity images. A cross-checking operation is performed and those "uncertain" pixels are detected and an interpolated disparity assignment is performed. This approach was evaluated using initial disparity from [14], showing significant improvement.

The method proposed in Mattoccia [23] is able to significantly reduce the execution time over [24], while achieving equivalent results. This improvement is due, in part, to this method's relaxed [21] and utilization of coarse-grained thread level parallelism.

According to Mattoccia [22, 25], the LC operation [21] can be further improved by "constraining its behavior on superpixels" using a mean-shift operation [7]. In this two phase strategy, the reference image is first over segmented, followed by a cross checking detection for "uncertain pixels," and "regularization" of disparities within the superpixels. Second, the segmentation constraints are relaxed to allow the disparities to propagate. As in the two previous examples, [22] uses the disparity image from [14] as a starting point.

### 1.2.2 Global-Based Approaches

Despite the success of many of the local-based approaches listed previously, global approaches, like graph cuts [6, 16, 18, 47, 4] and belief propagation [20, 28, 31, 30, 19, 5, 40], which utilize the entire image to facilitate information propagation in order to make an optimal match decision, continue to produce the top ranked results according to Middlebury [36]. Tappen and Freeman [41] conducted comparative tests using graph cuts and belief

propagation and concluded that overall they appeared evenly matched, but, while graph cuts produced smoother disparity images, belief propagation maintained the overall structure in a more acceptable manner. Further, both Sun et al. [39] and Nister et al. [49] state that belief propagation is a better choice for implementing on parallel hardware than graph cuts, while Nister et al. [49] also claims that belief propagation makes it easier to handle occlusion detection. For these reasons, especially the occlusion handling, this work uses belief propagation.

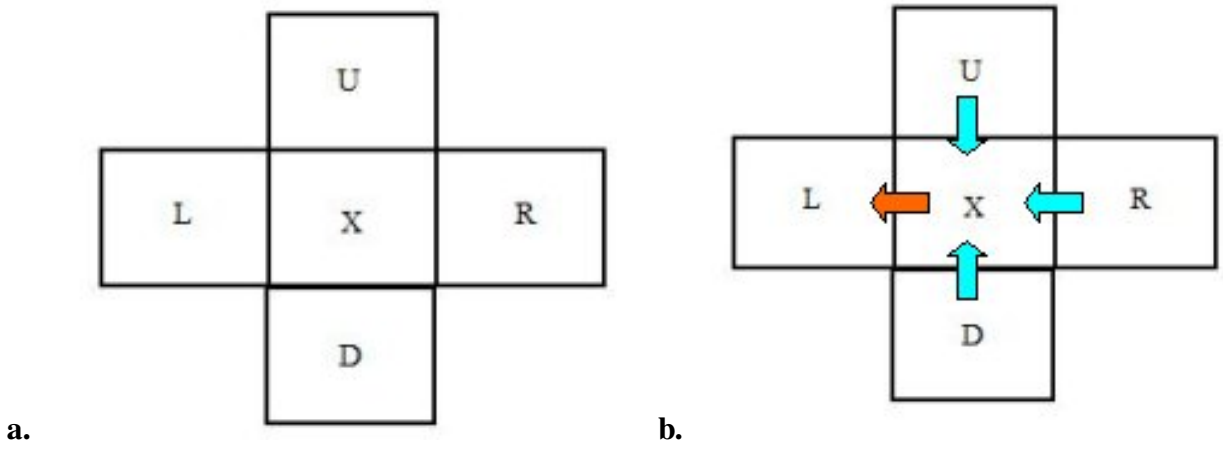



*Figure 1.1*: Message Grid; a. Sample Grid b. Example of incoming message use for outgoing message calculation

In standard belief propagation, it is assumed that each pixel in the reference image has a matching pixel in the search image, which can be found within a user-defined disparity range, say k. These disparities are assumed to be integers. Belief propagation is an iterative approach. During each iteration, each pixel sends messages to and receives messages from its four neighboring pixels (U, D, L, and R). When calculating the newest outgoing message for a particular neighbor, the current pixel uses its incoming messages from three of its four neighboring pixels, (i.e. all neighbors expect the one to which the current message is intended). For instance, if the current outgoing message from pixel X, shown in Figure

1.1b, is for the neighbor L, then only the incoming messages from D, U, and R will be used. Initially, all incoming messages are set to zero. In addition to the incoming messages, the current outgoing message also includes the data cost (typically calculated using the absolute difference between reference pixel and each pixel in its disparity range in the search image), and the smoothness cost. The smoothness cost is calculated as a penalty for discontinuous disparities between the current pixel and the intended recipient of the message. Otherwise, the smoothness cost is zero.

Once the iterations are complete, the final belief vector for each pixel is calculated by summing the incoming messages from each of the four neighboring pixels and the current pixel's data cost for each disparity in the range. The index of the minimum value in the belief vector is assigned as the final disparity at that reference pixel. In the standard belief propagation, a very large number of iterations are required to ensure convergence. Hence, the standard belief propagation is "too slow for practical use" [11].

Felzenszwalb and Huttenlocher [11] introduced an Efficient Belief Propagation approach using the concept of multiscale belief propagation, where a coarse-to-fine strategy is used. This process starts at the coarsest level, and solves a separate belief propagation problem at each level. Instead of coarsening the images themselves, as is done traditionally, Felzenszwalb and Huttenlocher coarsen only the data cost. This allows for fractional disparities at the coarser levels (with respect to the coarse pixel size). Once the propagation process is completed for the current level, its final outgoing messages are used to initialize the initial incoming messages for the next level. The initial incoming messages are set to zero only in the coarsest level. This initialization of the finer levels is what accelerates the overall convergence of the method. Felzenszwalb and Huttenlocher [11] also proposed an approach that reduces the smoothness cost computations from a $O(k^2)$ complexity to O(k). In addition, they introduced the concept of bipartite message updating, which updates only half the total number of pixels in the image during the current iteration. They have shown that their approach is much faster, while providing the same quality of results

as the standard implementations.

Other approaches have been recently developed utilizing the multiscale or hierarchical belief propagation approach to speedup convergence further. In Sarkis and Diepold [32], an additional speedup is achieved by reducing the number of pixels participating in belief propagation process, where the reference image is setup as a "sparse" network of "non-uniformly sampled" pixels using a "skewness" metric, with message updating restricted to the nodes in this network. Once the process is complete, the results within each node are used to interpolate the disparity values at the remaining pixels. Klaus et al. [17] sought to solve the problem of stereo matching by incorporating image segmentation and a "self-adapting dissimilarity measure" in their belief propagation. This dissimilarity measure was based on both the SAD and forward gradient-based measure for the surrounding neighborhood of the current pixel.

Xu and Jia [48] approached the problem of stereo matching based on the "confidence" of a given pixel being occluded. These confidence values are essential to the calculation of the data cost. The confidence calculation falls into one of three cases: (1) definitely occluded, (2) not occluded, or (3) possibly occluded. As these values must be between 0 and 1, the third case is often a fraction. The matching pixel falls within the first case if the "interframe consistency" between the reference and search image is violated, resulting in a confidence of 1. The third case is applied if the final belief of the current pixel x, $b_x(d^*)$, exceeds a user-defined threshold [48]. In case (3), the function, $T\left(\frac{b(d^*)-b_{min}}{b_o-b_{min}}\right)$, returns a value between 0 and 1 based on the parameter's value; $b_o$ is the average over all minimum beliefs found in case (1), and $b_{min}$ is the top user-defined percent of the minimum beliefs from all pixels. If neither of the previous cases is found to be true, case (2) sets the confidence to 0.

Trinh [44] incorporates hierarchical belief propagation and image segmentation, with the reference image being divided into segments or "superpixels." In order to improve efficiency, the messages are passed between neighboring segments, while maintaining smooth-

ness within the segment itself. However, unlike the standard multi-level version [11], Trinh's approach [44] had difficulties transferring message information between levels. Upon reaching a new level, the previous level's segments are further segmented, with each "child" segment initialized with a bias towards the final disparity of its parent segment. Additionally, the disparities of these "child" segments are often less than those of the parent, as they are estimated within a given range around the parent's value. Given this estimation, the results of the finer levels are less affected by errors in the coarse levels.

Nister et al. [49] updated the data term based on whether the reference pixel was occluded. The approach had three main step: (1) an initial stereo match, (2) classification of the pixels, and (3) an iterative refinement step. In step (1), a "correlation volume" based on a "color-weighted correlation" is chosen, as it is "less sensitive to occlusion boundaries" [49]. This step uses the data term calculated from the color-weight correlation in hierarchical belief propagation similar to [11]. This initial stereo is run for both images to facilitate a consistency check in the next step. In step (2), each pixel is labeled as occluded, stable or unstable, based on a set of tests. If the consistency check fails, the disparity value is classified as occluded. For those unoccluded pixels, the "correlation confidence" is measured against a user-defined threshold; if it exceeds the threshold, the pixel is considered stable. The goal for step (3) is to propagate information from stable pixels to all remaining pixels in the image. First, this step uses both image segmentation, implemented using mean shift [7], and plane fitting, implemented using RANSAC, to smooth the disparities at the stable pixels. The data terms are modified differently, with greater penalties for occluded and unstable pixels, as they need more "regularization" [49]. This step is intended to ensure that all occluded and unstable pixels are on the same plane as the stable pixels assigned to a given segment. The segment information, pixel classification, and correlation volume aid in calculating the new data term, which is used, in conjunction with the original smoothness term, in another hierarchical belief propagation.

In spite of the improvements made by approaches like [11] and the others mentioned

previously, few have achieved a real-time execution. However, Nister et al. [49, 50] developed a "fast converging" real-time belief propagation, which works well even with video. This approach achieved this impressive speedup by eliminating "redundancies" found in approaches like [11]. According to [49, 50], the main issue is ignoring previously converged pixels when updating messages or smoothness. If a pixel's incoming messages are found to be unchanged from a previous iteration, the operations for this pixel will be ignored in the future. The only exception to this rule is if a neighboring pixel's outgoing is updated at a later time.

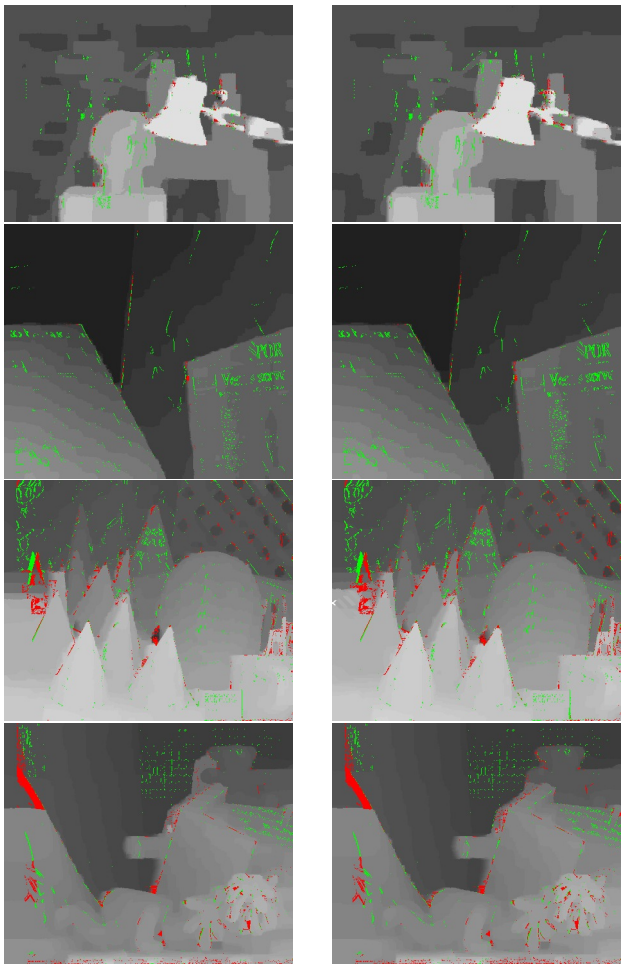


*Figure 1.2*: Comparison of large intensity differences between EBP images with different data cost methods: Absolute difference (left) versus Birchfield-Tomasi (right). Red pixels: bad matches and green pixels: good matches.

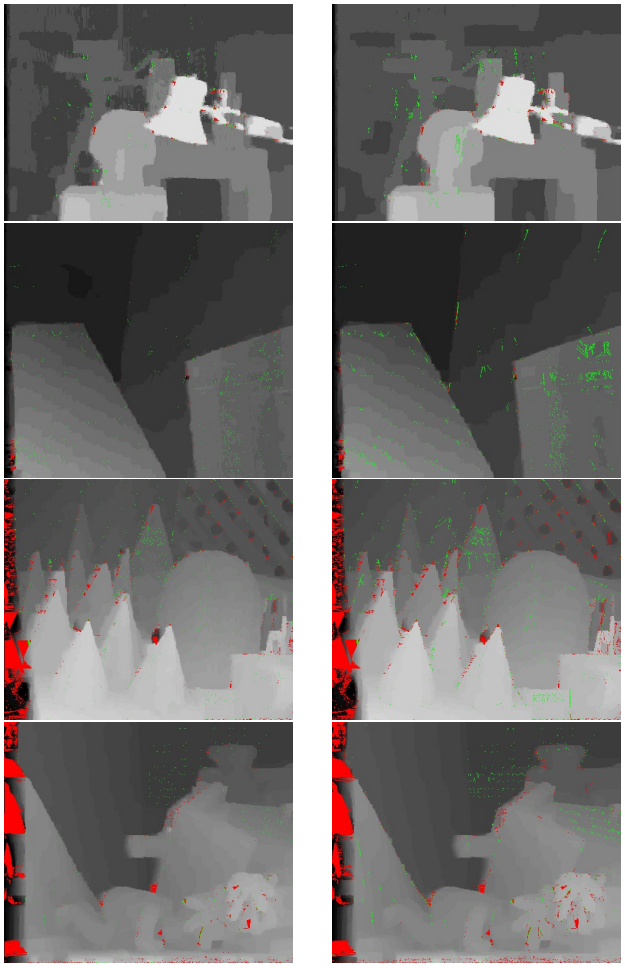


*Figure 1.3*: Comparison of large intensity differences between the modified EBP images with different data cost methods: Absolute difference (left) versus Birchfield-Tomasi (right). Red pixels: bad matches and green pixels: good matches.

While the true disparity values in the Middlebury dataset [36] are real numbers, most belief propagation approaches compute only integer disparity values. The disparity results using the code from Felzenszwalb and Huttenlocher [11] are shown in Figure 1.2. As shown in Figures 1.2 and 1.3, the green and red pixels, which represent good and bad matches, respectively, when compared with the ground truth, are where the intensity difference between the reference pixel and its corresponding match pixel in the search image is greater than a user-defined threshold; in this case, 20. In both figures, the intensity difference coloring is mostly found to be consistent along the occlusion boundaries within each

other images, as one would expect. In a few cases, there are other areas where the interior of an object also gets classified in the manner. The Efficient Belief Propagation results using both absolute difference data cost and Birchfield-Tomasi's pixel dissimilarity appear similar in their arrangement of large intensity difference matches. As one might expect, the modified Efficient Belief Propagation produces significantly fewer large intensity difference matches, due to its fractional correspondence. In this case, though both data cost calculations produce fewer large difference points, the modified EBP results using Birchfield-Tomasi are more similar to their standard Efficient Belief Propagation counterparts due to the symmetric nature of the algorithm. For instance, when examining correspondence from right to left, the best interval match maybe found in the left to right direction instead. For those pixels found in the smoothed interior of the object, further examination is necessary as these could be incorrect matches. Despite the success [11, 39, 32, 17, 49, 50, 48, 44], one has to ask "Does a smooth disparity image, such as Figure 1.2, reflect optimal matching results?" In this case, the term "optimal" may be defined as those results with a low error percentage when compared with the ground truth image.

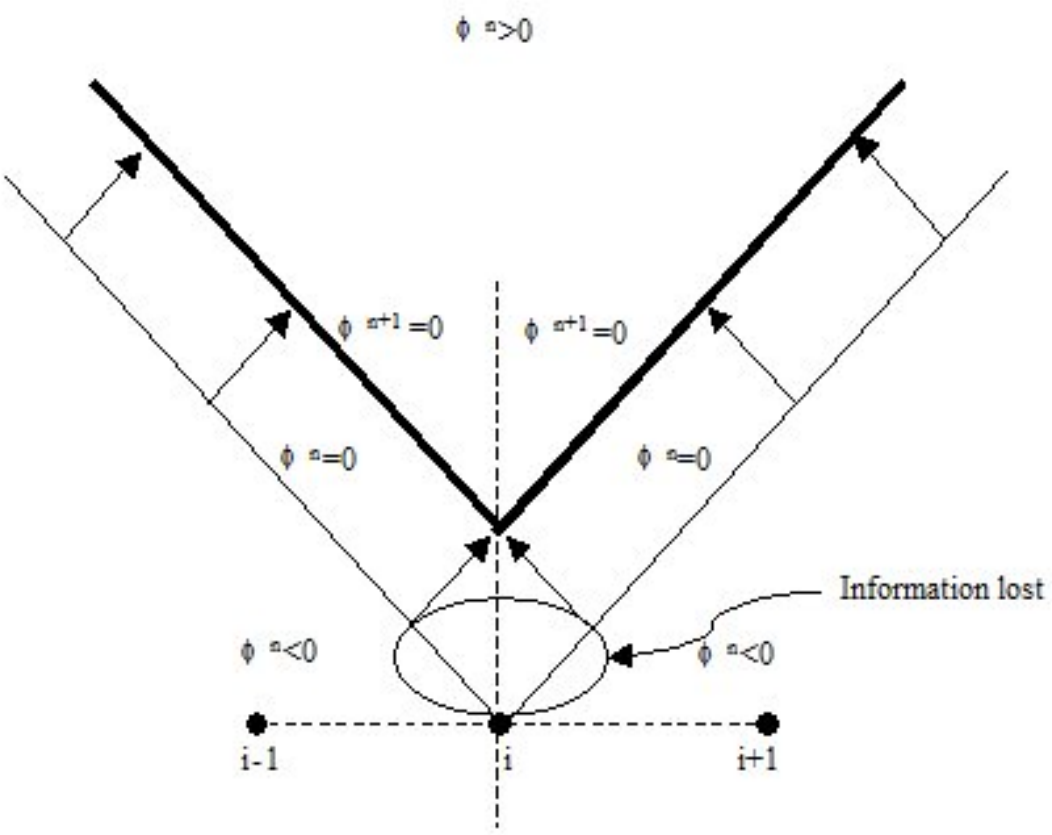
## 1.3 Level Set Method

In order to deal with such issues as the large intensity correspondences discussed in the previous section, concepts from the Level Set method (see Sethian [37]), discussed in the section 2.1, are incorporated into belief propagation stereo matching in the present work. The objective of the Level Set method is to estimate the motion of a "front" within a given space [37], where the motion is governed according to some specified speed function. In order to estimate this motion, the challenge of modeling the speed function must be handled independently [37]. According to Sethian [37], the speed function (F) is defined as F= F(L,G,I), where L is the local properties, G is the global properties of the front, and I is the independent properties. The local properties are those that include the local geometric information, like curvature and normal direction. While the global properties depend on

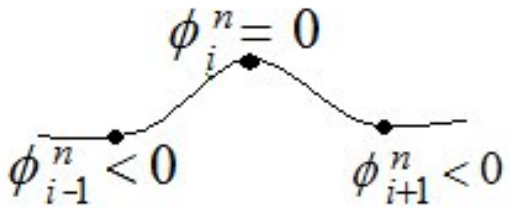
the shape and position of the given front, the independent properties are not dependent on the shape of the front, instead deal with the underlying fluid velocity that moves the front.

The value of the speed function is instrumental in determining the motion of the front. For instance, if the area has a positive uniform speed (F=1), the motion of the front will always be uniformly outward. The level set method is also setup to utilize an adaptive speed function, which allows the front to move both forward and backward.

During the motion of a given front, its curvature will often be smooth. However, these smooth curves can quickly form "non-differentiable" corners, as shown in Figures 1.4 and 1.5, which necessitates a "weak solution." One such solution is based on the "entropy condition," which can be understood in analogy with the motion of a forest fire. With respect to the forest fire, the entropy condition states that once a point has been "burnt" or visited, it cannot be revisited. According to Sethian [37], any attempt to do so will not allow for regaining of previous information, as it is lost.

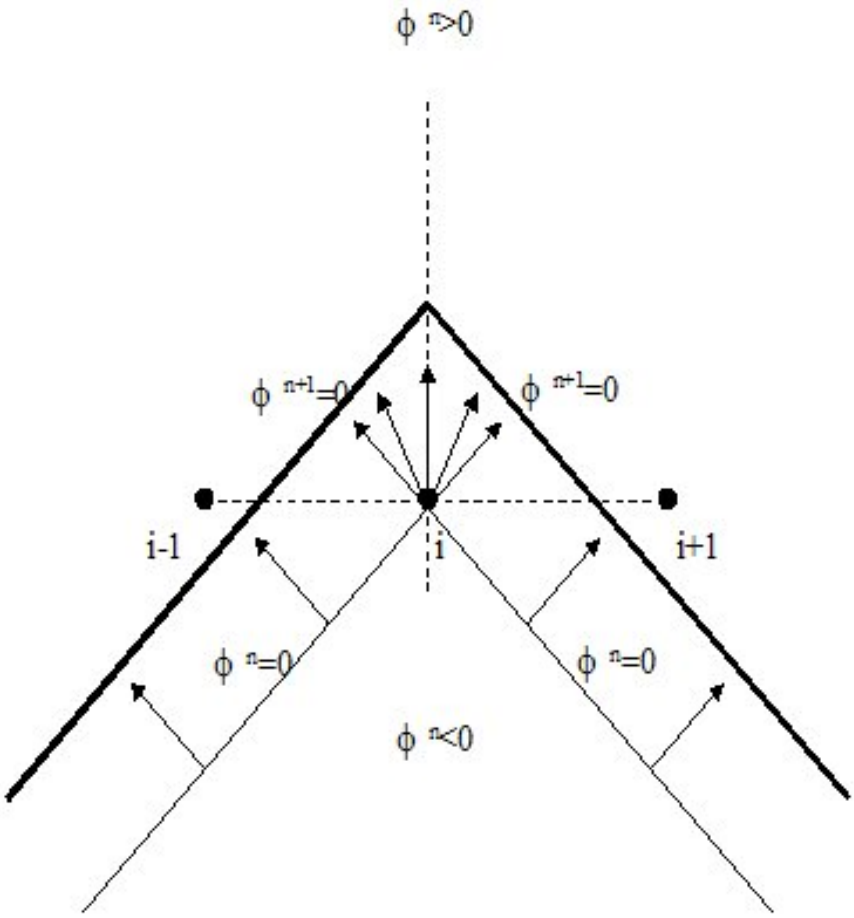



a. Shocks result in loss of information: The dotted lines are grid lines and the solid dots are grid points. The thin solid line is the zero level set at time n, $\phi^n = 0$. Arrows indicate the direction of motion. The bold solid line is the new zero level set at time n+1, $\phi^{n+1} = 0$. The motion from time n to time n+1 results in the loss of information.

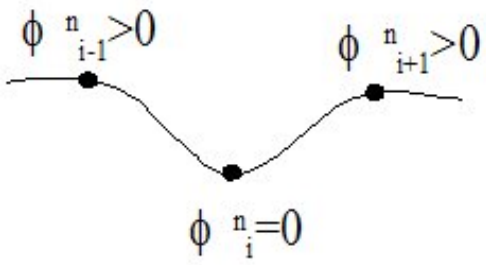



b. Shocks occur at local maximum for the above motion: The horizontal profile of $\phi^n$ at grid point i showing a local maximum at i.

*Figure 1.4*: First Classification for Front Propagation: Shock.




a. Rarefaction: The dotted lines are grid lines and the solid dots are grid points. The thin solid line is the zero level set at time n, $\phi^n = 0$. Arrows indicate the direction of motion. The bold solid line is the new zero level set at time n+1, $\phi^{n+1} = 0$. Unlike shock, the motion from time n to time n+1 does not result in the loss of information.




b. Rarefactions occur at local minimum for the above motion: The horizontal profile of $\phi^n$ at grid point i showing a local minimum at i.

*Figure 1.5*: Second Classification for Front Propagation: Rarefaction.

The cases shown in Figures 1.4 and 1.5 are regulated according to the speed values for both the left and right portion of the front [37]. The arrows on either side of the front represent the direction of motion. The evolution of the front in time is solved though the development of a solution to the local Riemann problem, as seen in Figure 1.6 [37].

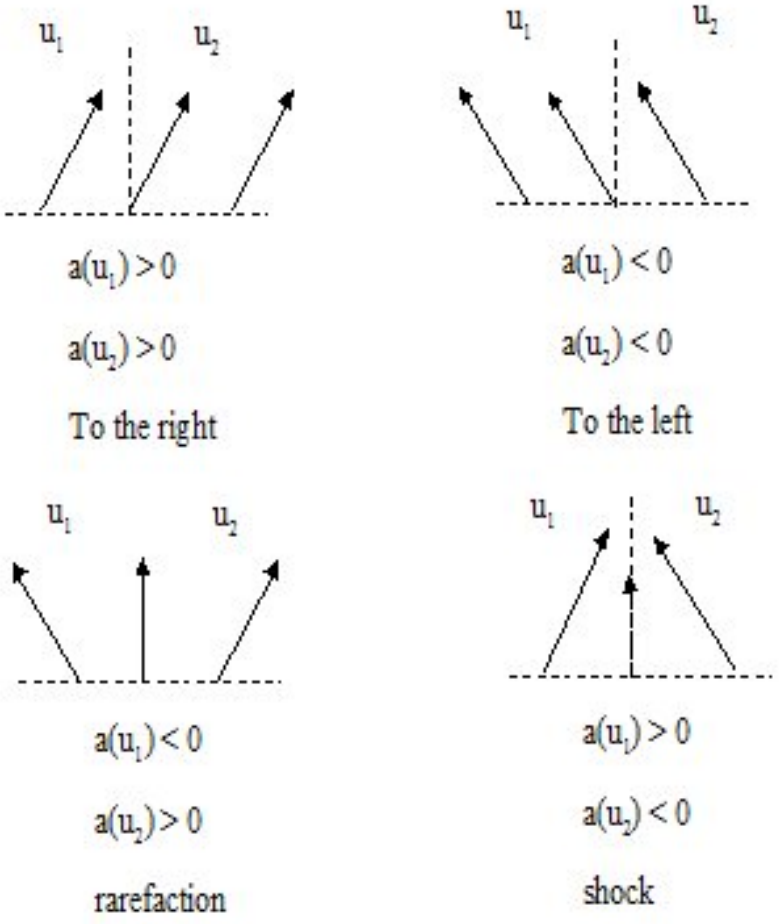



*Figure 1.6*: Possible solutions to local Riemann problem. a(u1) and a(u2) are the speed functions for the states u1 and u2.

As originally shown by Engquist and Osher [10], when solving non-linear hyperbolic equations, if the speed function to the left of a point is negative and the speed function to the right is positive, the motion will result in a "rarefaction fan" at that point, as seen in

Figure 1.4 and Figure 1.6c. On the other hand, if the speed function to the left is positive and the right is negative, the motion will result in a "shockwave," as seen in Figure 1.4 and Figure 1.6d.

The rest of this work is organized as follows. The next section presents an overview of the current approach, explaining how ideas from level set are incorporated into belief propagation. Specifically, the proposed approach includes three new modifications to the Efficient Belief Propagation approach [11]. They are (1) modification to the initial data cost to encourage reference pixels to match to an interval in its disparity range rather than to a pixel, (2) mutual consistency adjustment to encourage matches between extremal pixels of the same type in both images, and (3) shifting of the minimum disparity within outgoing messages based on the modified initial data cost of the two pixels involved to provide a balance for the smoothness term. The operation in (1) is useful especially when corresponding fronts in the reference and search images have an unequal number of pixels. The following section presents results of the current work. The last section draws conclusions and presents direction for future work.

# Chapter 2

# MODIFIED EFFICIENT BELIEF PROPAGATION- PRESENT WORK

## 2.1 Framework Using Level Set Concepts

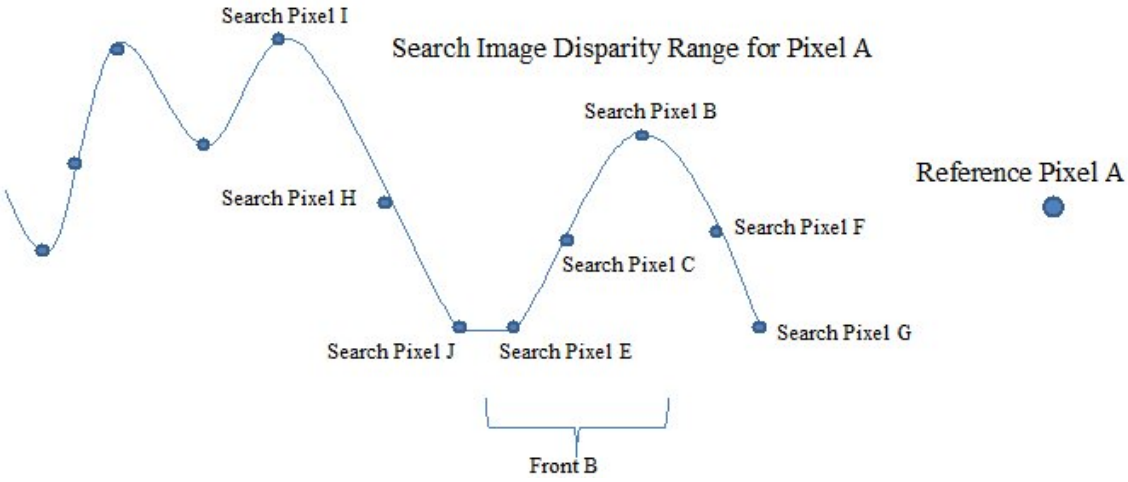



*Figure 2.1*: Level Set Framework. The wave structure shown above indicates the varying pixel intensity values along the current scanline. Each pixel is classified as either a local extrema or local non-extrema based on the product of the forward and backward difference between the intensity values of the current pixel and its left and right neighbors. Two neighboring extremal points, such as pixels B and E, form a feature front. In the case of front B, shown above, these two points are classified as a max and min extremal point respectively. Each front consists of both a front type and strength; both are based on the intensity differences between the extremal endpoints. Extremal points, like those listed above, as well as pixels I and J, belong to two fronts; retaining the information concerning both fronts in which they are associated, as well as a strength value based on the maximum strength of their two fronts. On the other hand, non-extremal points, like pixels C and F-H, are assigned both a front type and strength based on the front in which they are located.

*Table 2.1*: Level Set Extremal and Non-Extremal Pixels

| | | Number of Pts. | Percent of Total | Strong Pts. (PoT) | Strong Pts. (PoPts) |
|---|---|---|---|---|---|
| Tsukuba | EX | 73147 | 66.1413 | 26.1447 | 39.5286 |
| | NE | 37445 | 33.38587 | 28.1883 | 83.2528 |
| Venus | EX | 104107 | 62.6313 | 27.0614 | 43.2075 |
| | NE | 62115 | 37.3687 | 24.3752 | 65.229 |
| Cones | EX | 91686 | 54.3324 | 39.6486 | 72.9741 |
| | NE | 77064 | 45.6676 | 40.1333 | 87.8815 |
| Teddy | EX | 96837 | 57.3849 | 32.1132 | 55.961 |
| | NE | 71913 | 42.6151 | 32.9079 | 77.2211 |

According to the Level Set theory, shocks and rarefactions, as shown in Figures 1.4 and 1.5, occur at local extrema. In this new framework, each pixel in the reference and search images is assigned a type classification: extremal (min or max), or non-extremal. Forward and backward differences are used for classifying a pixel. For instance, if the backward difference is negative and the forward difference is positive, the pixel is classified as a min. On the other hand, if the backward difference is positive and the forward difference is negative, the pixel is classified as a max. However, if one of these intensity differences is 0, the pixel is classified as a flat min or flat max, based on the other difference. All remaining pixels are classified as non-extremal pixels. As shown in Figure 2.1, the wave structure can be divided into fronts, based on neighboring extrema. Each extremal pixel is associated with two different fronts; both the type and strength of each front are calculated using the intensity differences of these extremal pixels. Fronts are classified in three categories: ascending, descending, or flat, based on the signed intensity difference of the neighboring extremal pixels on a give scanline traveling from left to right. For instance, if the backward difference for the current extremal pixel is negative, the left front associated with this pixel is classified as descending; if positive, ascending. The forward difference applies the opposite classification criteria for the right front; positive is ascending and negative is descending. All remaining fronts are classified as flat. The absolute forward and backward differences are used to calculate the strengths of the left and right fronts of the current ex-

tremal pixel, and the strength at the extrema is defined based on the maximum of its left and right strengths. All non-extremal pixels are assigned a strength and front classification based on the front to which they belong.

Table 2.1 shows the number of extremal and non-extremal pixels that are found in each of the input images, as well as the overall percentage of the image in which these pixels represent. Additionally, most of the object within each of the images are surrounded to the left and right on a given scanline by "strong fronts," meaning those with a strength greater than a user-defined threshold, in this case 5. In the last two columns in Table 2.1, the percentage of total (PoT) shows how many strong pixels, with a strength greater than 5, are found within the entire image, while the percentage of points (PoPts) shows the percentage of strong points from within the extrema and non-extremal groups.

From Table 2.1, several facts about local extrema versus non-extrema can be deduced. For instance, there are typically more local extrema than non-extrema. This may come as a surprise, as each front only has two extremal pixels, but often numerous non-extrema. This can be explained by the fact that within the low textured regions, consisting of weaker fronts, the majority of the pixels are located on fronts consisting of only two pixels and as classified as local extrema. In most cases, the local non-extrema found in these areas are either located on a flat-type front or a very weak front with only two or three non-extrema. Additionally, the largest portion of both extrema and non-extrema are classified as "weak," due to their low strength, especially those found in low texture regions. While the "strong" points represent a low percentage of all the pixels in the image, a significant majority of the extrema and non-extrema are classified as "strong."

## 2.2 Modified Initial Data Cost

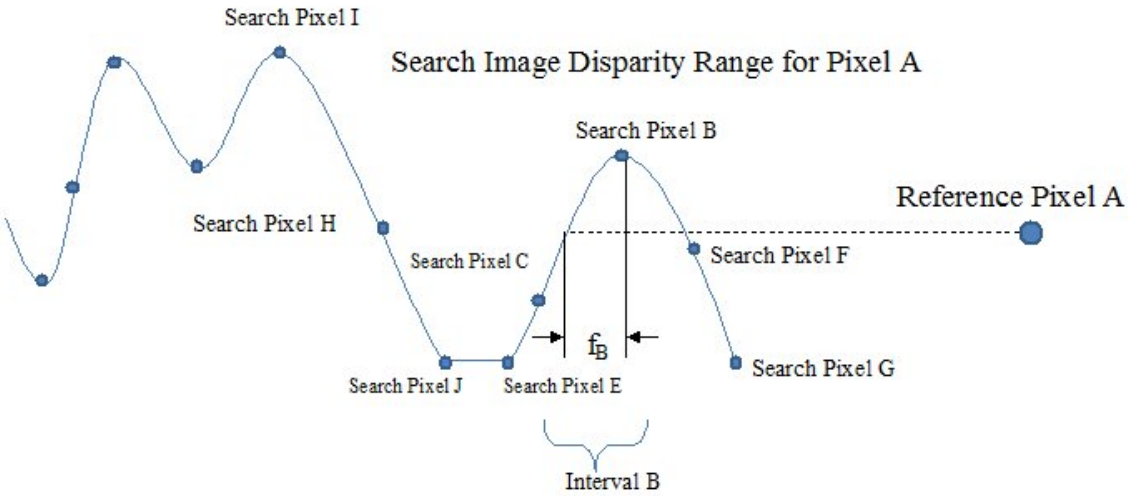



*Figure 2.2*: Modified Initial Data Cost. An interval is formed by consecutive pixels such as interval B. This illustration shows that the intensity at pixel A intersects interval B; a fractional disparity ($f_B$) exists within this interval. Additionally, the data cost for pixel B is set to 0, as it represents this interval. This procedure is repeated at all intervals in pixel A's disparity range.

In the results, shown in the next chapter, the data cost is initially calculated using two methods: the standard absolute difference as well as the work of Birchfield and Tomasi [2], for a comparison. However, in cases like interval B in Figure 2.2, a modified initial data cost is calculated as follows: as shown in Figure 2.2, an interval is defined as two neighboring pixels on the same scanline and the starting point of each interval is referenced according to an integer disparity value. For instance, interval B in Figure 2.2 is referenced with respect to the search image pixel B, where each interval represents an intensity range. If the reference pixel's intensity falls within this range, a fractional disparity will be set for the interval. The fractional disparity $f_B$ is calculated as $f_B = d_B + \frac{|I_B^L - I_A^R|}{|I_C^L - I_B^L|}$ , where $d_B$ is the integer disparity at B, $I_B^L$ and $I_C^L$ are the intensity values on the search image scanline, and $I_A^R$ is the intensity value for the current pixel on the reference scanline. Additionally, the

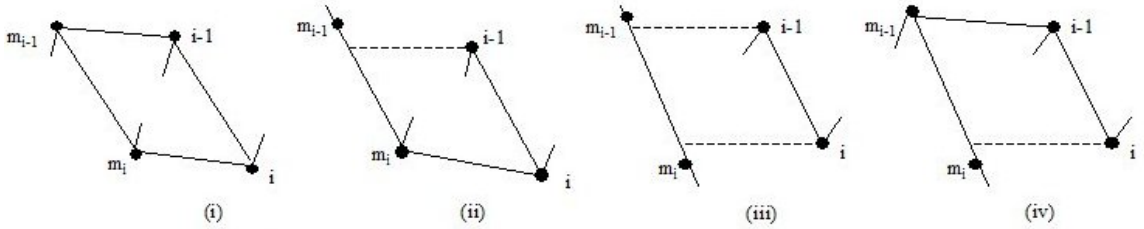
data cost at this disparity value is set to 0. This concept of intervals is especially helpful in those instances where corresponding fronts in the reference and search images have an unequal number of non-extremal pixels.

## 2.3 Message Adjustment Based on Zero Data Costs

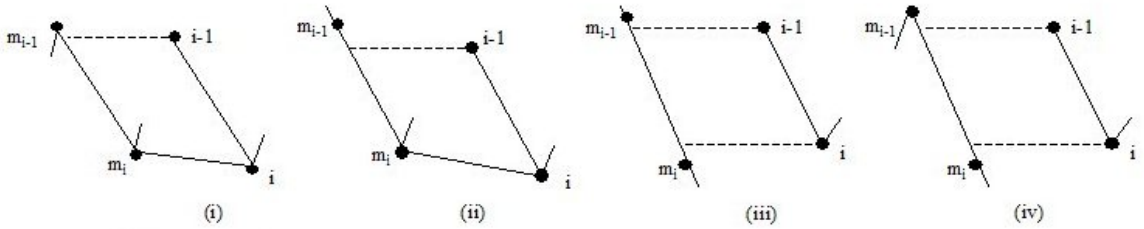
As stated previously, belief propagation includes a smoothness cost penalty in each outgoing message for discontinuity between the sender and receiver. In addition to the standard penalty given by the discontinuity difference, the smoothness cost is regulated by a constant value known as the "rate of increase." This rate of increase allows each message to relay the amount of discontinuity that will be tolerated with higher values resulting in more smoothed, continuous disparity regions. In belief propagation, this smoothness cost will force each pixel to seek a unique solution often in-line with the initial correspondence suggested by the data cost.

However, mistakes are a result of sampling related errors in pixel distribution. The operation discussed in this section provides a balance to the smoothness cost by allowing for an interval-based alternative correspondence to be located.

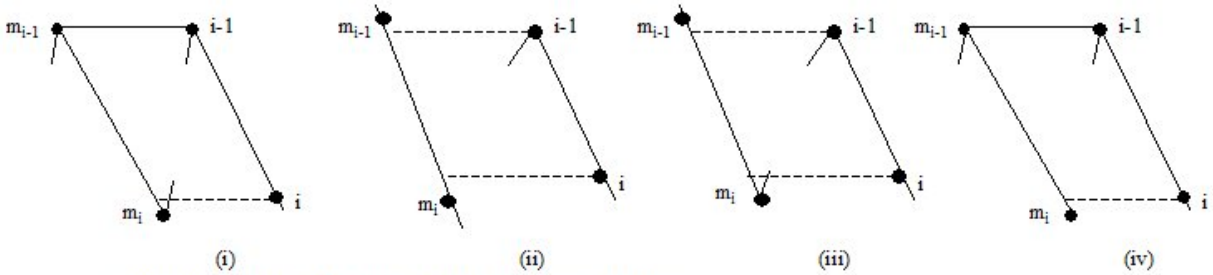
This adjustment is intended to shift the minimum cost location in the outgoing message since the final disparity is affected by the minimum cost location. If this sender is a non-extrema, this adjustment is applied if the data cost is 0 at the minimum cost interval of the current pixel (i)'s outgoing message. On the other hand, if the sender is an extrema, a data cost of 0 is not required if the sender's match is an extrema of the same type as the sender. For the purposes of this discussion, assume that the neighbor receiving the message is the left neighbor (i-1). The outgoing messages can be categorized into four cases based on the type of the sender and receiver: (1) extrema sends to extrema, (2) extrema sends to non-extrema, (3) non-extrema sends to extrema, and (4) non-extrema sends to non-extrema. All cases require that the fronts in both the left and right image be of the same type. These cases are treated as follows; refer to Figure 2.3.



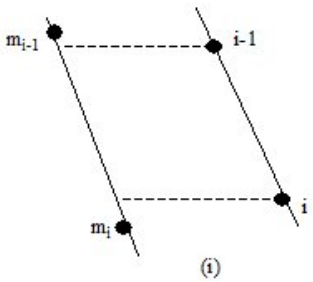

a. Case 1: Sender and Receiver are Extrema




b. Case 2: Sender is an Extrema and Receiver is a Non-extrema




c. Case 3: Sender is a Non-extrema and Receiver is an Extrema




d. Case 4: Sender is a Non-extrema and Receiver is a Non-extrema

*Figure 2.3*: Message Adjustment Based on Zero Data Cost ( i [current reference pixel], i-1 [left neighbor of reference pixel], $m_i$ [match for current reference pixel], and $m_{i-1}$ [match for left neighbor] ).

Case 1: Both the Sender i and Receiver i-1 are extrema
(a) If $m_i$ is of the same extrema type as i and $m_{i-1}$ is of the same extrema type as i-1, then no shifting is required. Figure 2.3a (i)
(b) If i is of the same extrema type as $m_i$, but $m_{i-1}$ is a non-extrema, then check the interval at $m_i$ to see if i-1 has a data cost of 0. If it does, shift the minimum for i's outgoing message by -1. Figure 2.3a (ii)
(c) If $m_i$ and $m_{i-1}$ are both non-extrema pixels, then check the interval at $m_i$ to see if both i and i-1 have a data cost of 0. If they do, shift the minimum for i's outgoing message. Figure 2.3a (iii)
(d) If $m_{i-1}$ is of the same extrema type as i-1 and i has a data cost of 0 for the interval at $m_i$, then no shifting is required. Figure 2.3a (iv)

Case 2: Sender i is an extrema* and Receiver i-1 is a non-extrema
(a) If i is of the same extrema type as $m_i$ and i-1 has a data cost of 0 for the interval at $m_i$, then shift the minimum for i's outgoing message. Figure 2.3b (i)
(b) If $m_i$ is a non-extrema, and both i and i-1 have a data cost of 0 for the interval at $m_i$, then shift the minimum for i's outgoing message. Figure 2.3b (ii)
(c, d) If both i and i-1 have a data cost of 0 for the interval at $m_i$, then shift the minimum for i's outgoing message. Figure 2.3b (iii, iv)
* Both extrema types (min and max) are handled in same manner.

Case 3: Sender is a non-extrema and Receiver is an extrema
(a , d) If $m_{i-1}$ is of the same extrema type as i-1 and i has a data cost of 0 for the interval at $m_i$, then no shifting is required. Figure 2.3c (i, iv)
(b) If $m_{i-1}$ is a non-extrema and both i and i-1 have a data cost of 0 for the interval at $m_i$, then shift the minimum for i's outgoing message. Figure 2.3c (ii)
(c) If both i and i-1 have a data cost of 0 for the interval at $m_i$, then shift the minimum for i's outgoing message. Figure 2.3c (iii)

Case 4: Both the Sender and Receiver are non-extrema
(a) If both i and i-1 have a data cost of 0 for the interval at $m_i$, then shift the minimum for i's outgoing message. Figure 2.3d (i)

** In all the above cases, $m_i$ and $m_{i-1}$ represent the initial match points in the reference image for i and i-1, respectively.
*** Additionally, right-to-left correspondence includes the interval $[I_{m_i}, I_{m_{i-1}}]$ in the search image for determining zero data cost, while left-to-right correspondence has a similar interval setup $[I_{m_i}, I_{m_{i+1}}]$.

This shifting operation moves the minimum cost location for i's outgoing message one point to the right. The operation is only applied to the left and right outgoing messages, and only when both i and i-1 meet the minimum strength threshold. The outgoing message for

the right neighbor is handled in the same way, except that the minimum disparity is shifted to the left.

## 2.4 Mutual Correspondence Adjustment

This adjustment is intended to encourage mutual correspondence between extremal pixels in the left and right images by adjusting the minimum cost locations for the respective outgoing messages. This mutual correspondence seeks to match pixels between the left and right images, where the match pixel is of the same type as the current pixel (i). When adjustment applies only in those cases where the sender is an extremal pixel, the type of the receiver is irrelevant. Mutual correspondence requires examining the corresponding scanlines from the left and right images simultaneously.

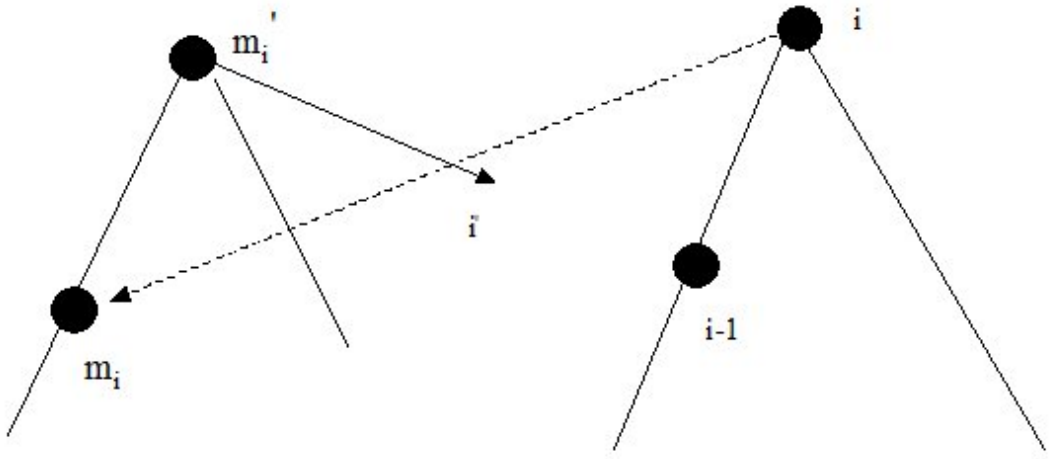



*Figure 2.4*: Profile Example for Mutual Consistency

Consider the illustration in Figure 2.4, where pixel i (the current reference pixel) matches pixel $m_i$ in the search image. If pixels i and $m_i$ are of the same type or i has a data cost of 0 for the interval at $m_i$, no adjustment is required. Otherwise, a 1-pixel neighborhood of $m_i$ on the left scanline is searched for a pixel of the same type as i; if none is found, no

adjustment is made. Suppose one of the 1-pixel neighborhood, say $m_i'$, is the same type as i. Then, the match for $m_i'$ on the right scanline ($i'$) is found; if $i'$ is not within the 1-pixel neighborhood of i, then no adjustment is made. If $m_i'$ matches i, then only minimum cost location for i's outgoing message is reset to point to $m_i'$. Otherwise, if the intensity differences between (i and $i'$) and ($m_i$ and $m_i'$) is significant, the minimum cost locations for the outgoing messages from $m_i'$ and i are adjusted; the new minimum cost location for both messages is the pixel difference between i and $m_i'$.

# Chapter 3

# MODIFIED EFFICIENT BELIEF PROPAGATION- RESULTS

For all the experiments discussed in this chapter, each algorithm, using the input image shown in Figure 3.1, was run on a Pentium D Dual Core 2.8 GHz computer. The algorithms do not use any truncation for data cost or messages or normalization of intensities for the data cost. For each algorithm, five coarse-to-fine levels were used and each level had five iterations. While the new modified approach performed several additional operations, it retained all the elements found in the standard EBP. As for the parameters, given that truncation was removed when calculating messages, the parameter 'd' used in the original EBP approach was not necessary. However, the rate of increase, 's', used when calculating the smoothness cost was set to 10.0.

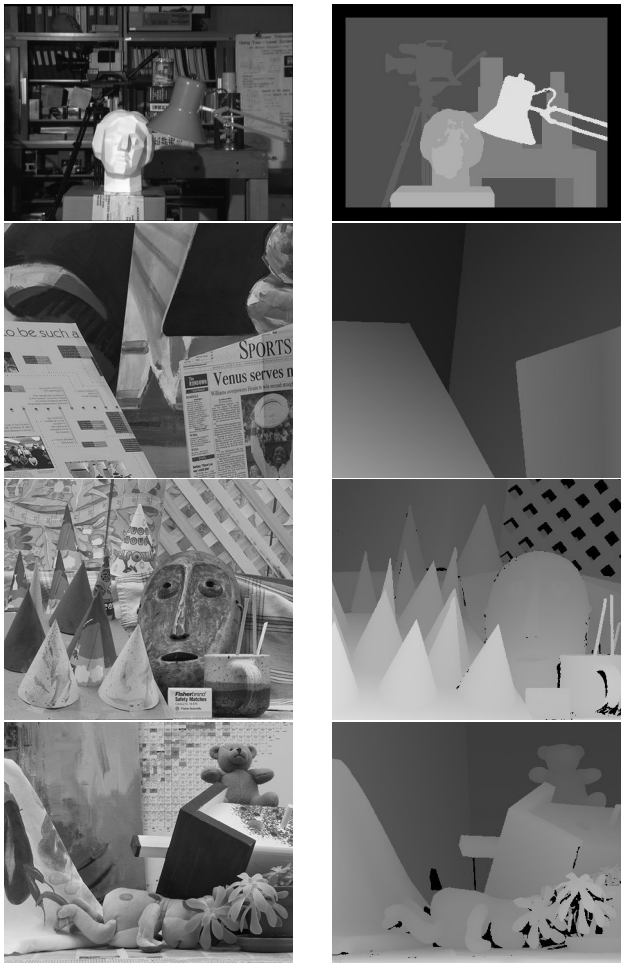



*Figure 3.1*: Images: Reference image (left column) and ground truth image (right column) produced by Middlebury [34, 33, 35, 15].

In Figures 3.2 - 3.5, the disparity results show a comparison between Efficient Belief Propagation and the modified Efficient Belief Propagation using both absolute difference and Birchfield-Tomasi's pixel dissimilarity to calculate the initial data cost. The first point to mention is that the results produced in this work are similar to those of the Efficient Belief Propagation. In the Efficient Belief Propagation results, the Birchfield-Tomasi results appear to be smoother than those produced using the absolute difference, often resulting in fewer errors when compared with the ground truth. For instance, less discontinuity is shown in all the images than in the absolute difference results. Additionally, BT does not produce as much "bleeding" of disparity values between objects as seen in the results of

AD; for instance, EBP's Teddy results in Figure 3.2. On the other hand, the smoothness of BT often sacrifices some of the sharpness of the objects' edges in images like Cones and Teddy that is found in the absolute difference results.

This occurrence is also found in these same images when used as input for the modified Efficient Belief Propagation. However, in this case, both absolute difference and Birchfield-Tomasi provide a slight improvement over their Efficient Belief Propagation counterparts. Though this difference is seen less in the case of BT, as the "cross-checking" interval process in this algorithm may often interfere with the shifting operations in mEBP. One example of the difference in the operations between EBP and mEBP is the manner in which to handle out-of-bounds disparity locations when calculating the data cost. As one may notice, the standard approach is to set all data cost within the right or left edge disparity region, depending on the direction of motion between the current images, to 0 and place the final disparity decision solely on the messages sent to these pixels by the neighboring pixels. Despite the smoother appearance along the edges, this process often causes a loss of information, as shown in Figures 3.2 and 3.3 for Cones and Teddy, where portions or entire objects are lost due to the strong influence of neighboring points with different disparity values and the lack of an influence from data cost of the current pixel. On the other hand, the modified version provides some autonomy by calculating all disparity values within the bounds of the image and restricting those that are not, allowing each pixel to calculate its data cost and choose a final disparity from all available locations, while discouraging choosing any location outside the image.

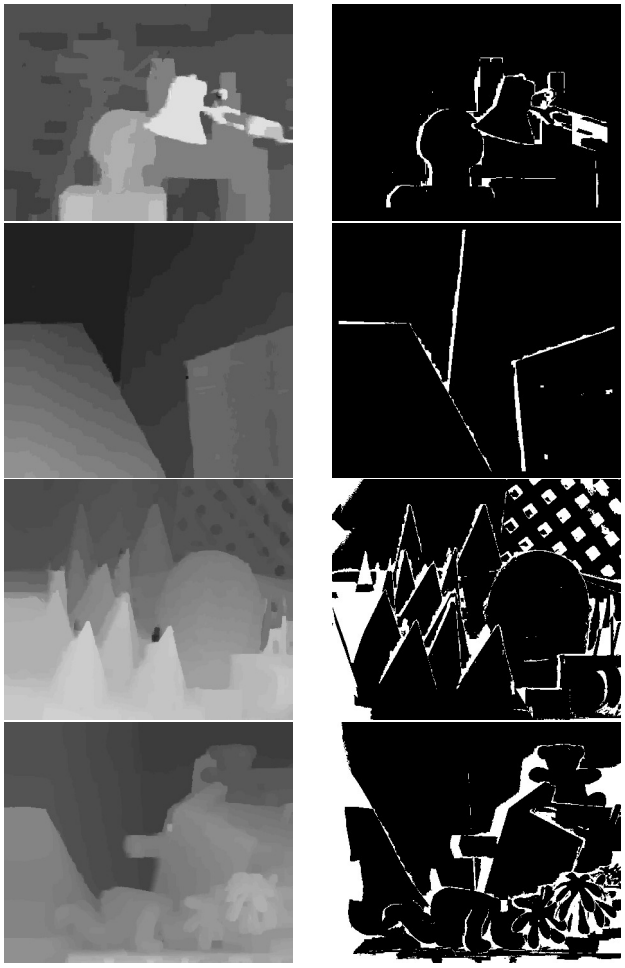


*Figure 3.2*: Comparison of EBP images with different data cost methods: Absolute difference. Disparity results (left) and Bad pixels (error>1.0) (right).

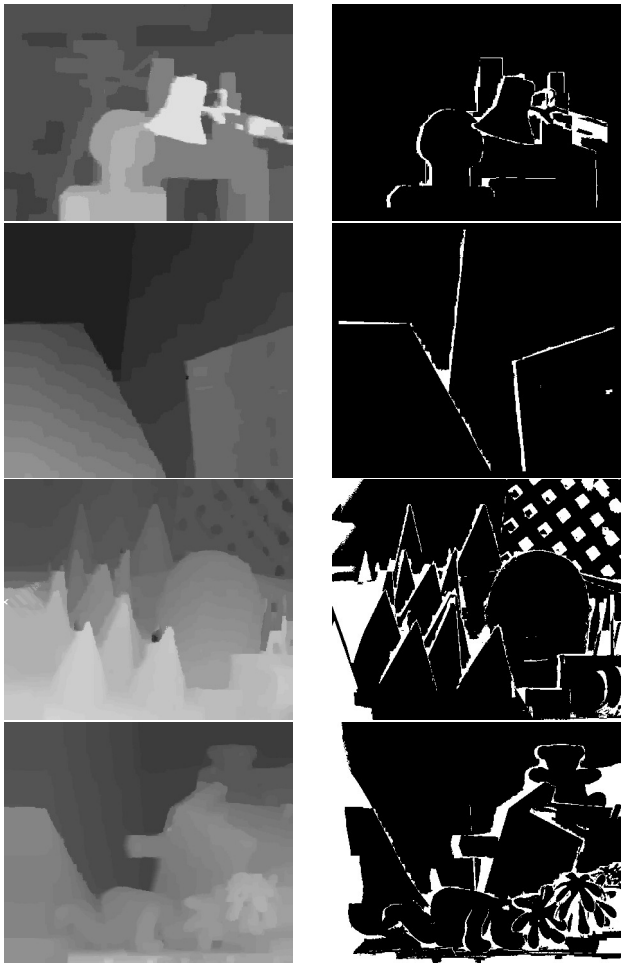


*Figure 3.3*: Comparison of EBP images with different data cost methods: Birchfield-Tomasi. Disparity results (left) and Bad pixels (error>1.0) (right).

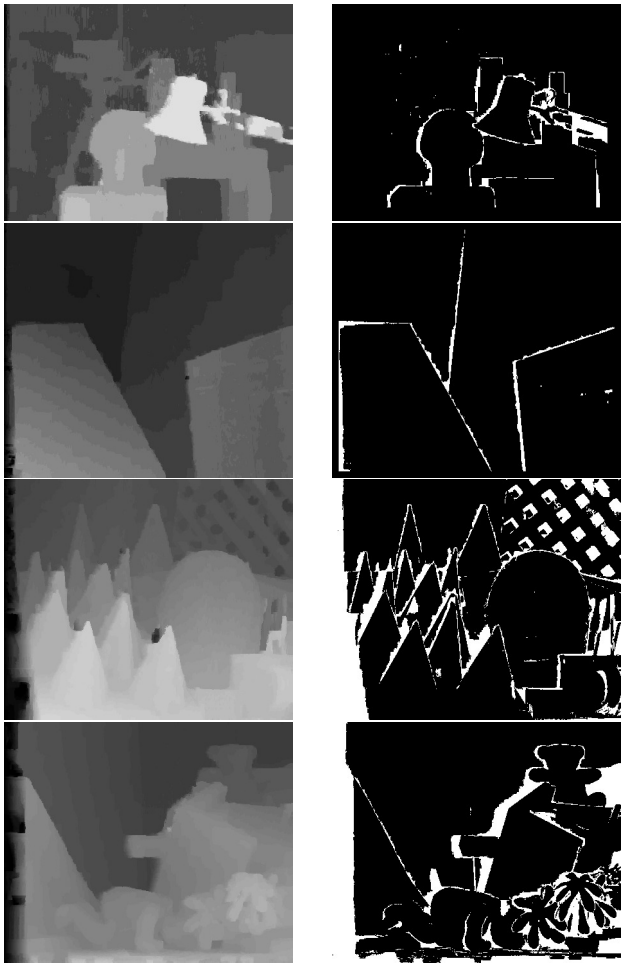


*Figure 3.4*: Comparison of modified EBP images with different data cost methods: Absolute difference. Disparity results (left) and Bad pixels (error>1.0) (right).

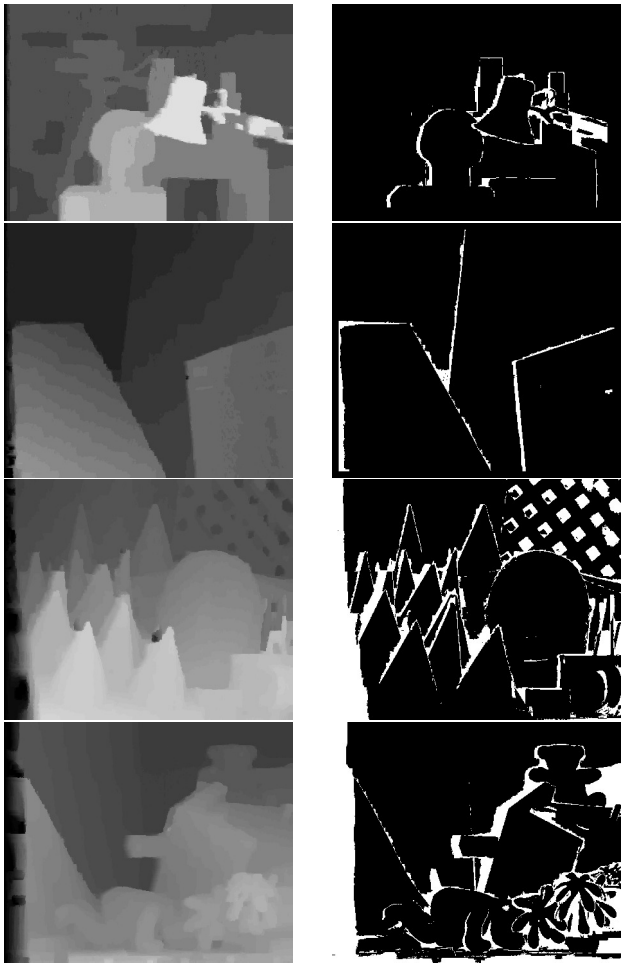


*Figure 3.5*: Comparison of modified EBP images with different data cost methods: Birchfield-Tomasi. Disparity results (left) and Bad pixels (error>1.0) (right).

*Table 3.1*: Execution Times (sec)

| | | Tsukuba | | Venus | | Cones | | Teddy | |
|---|---|---|---|---|---|---|---|---|---|
| | | AD | BT | AD | BT | AD | BT | AD | BT |
| EBP | MM | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | IDC | 0 | 0 | 0 | 2 | 1 | 3 | 0 | 3 |
| | HBP | 2 | 1 | 3 | 4 | 9 | 7 | 7 | 7 |
| mEBP | MM | 1 | 0 | 1 | 2 | 1 | 2 | 1 | 3 |
| | IDC | 0 | 1 | 0 | 2 | 1 | 4 | 1 | 4 |
| | HBP | 3 | 4 | 8 | 8 | 12 | 12 | 11 | 11 |

In Table 3.1, the time calculations show what one would expect; the standard EBP is

obviously faster, as it has fewer operations. However, one point of interest is that even with the increased computation requirement, the modified version still required, in most cases, less than 15 seconds to run. Additionally, one can see that the overall increase in the computation time for each image was often proportional to the length of its disparity range, with the message passing operations in belief propagation requiring the most time.

*Table 3.2*: Middlebury Results (Non-Occluded Error Percentage and Overall Average Ranks)

| | Rank | | Tsukuba | | Venus | | Teddy | | Cones | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AD | BT | AD | BT | AD | BT | AD | BT | AD | BT |
| EBP | 72.0 | 71.5 | 2.85 | 2.85 | 1.41 | 1.22 | 13.7 | 13.0 | 8.82 | 8.69 |
| mEBP | 71.2 | 70.5 | 3.09 | 2.83 | 1.35 | 1.54 | 10.4 | 10.3 | 8.47 | 6.73 |

*Table 3.3*: Middlebury Rankings for Non-Occluded Pixels

| | | Tsukuba | | Venus | | Teddy | | Cones | |
|---|---|---|---|---|---|---|---|---|---|
| | | AD | BT | AD | BT | AD | BT | AD | BT |
| Rank | EBP | 57 | 58 | 62 | 57 | 78 | 78 | 76 | 76 |
| | mEBP | 64 | 57 | 61 | 64 | 66 | 66 | 76 | 72 |

In Tables 3.2 and 3.3, the accuracy ratings from the Middlebury Evaluation for non-occluded pixels are shown for all the above images. Though the rankings for both EBP and mEBP are lower than those of other implementations, the accuracy of the modified approach is slightly higher for both absolute difference and Birchfield-Tomasi[2]. In those instances where the standard EBP has a lower error percentage, the difference is minimal. As for the rankings, many of the top approaches, especially the global ones, use additional operations like image segmentations as well as special formulas for identifying and handling occlusion. This new approach does not perform any of these additional operations, because, like Felzenszwalb and Huttenlocher [11], it is intended to provide a simple solution, while showing a slight improvement in accuracy through the use of levelset-based operations. This first set of Middlebury results only outlines the error percentages of the

non-occluded pixels as neither method currently has an explicit means of handling occlusion.

*Table 3.4*: Middlebury Results (Occluded and Discontinuous Error Percentage and Average Overall Error Percentage)

| | | | Tsukuba | | | | Venus | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Average Error | | AD | | BT | | AD | | BT | |
| | AD | BT | Occ | Discon | Occ | Discon | Occ | Discon | Occ | Discon |
| EBP | 12.8 | 12.5 | 4.94 | 15.0 | 4.88 | 15.1 | 2.41 | 18.2 | 2.22 | 15.4 |
| mEBP | 12.1 | 11.8 | 5.01 | 16.2 | 4.84 | 15.0 | 2.98 | 16.4 | 3.21 | 18.6 |

*Table 3.5*: Middlebury Results (Occluded and Discontinuous Error Percentages)

| | Teddy | | | | Cones | | | |
|---|---|---|---|---|---|---|---|---|
| | AD | | BT | | AD | | BT | |
| | Occ | Discon | Occ | Discon | Occ | Discon | Occ | Discon |
| EBP | 22.6 | 27.1 | 22.0 | 27.1 | 17.3 | 19.4 | 17.2 | 20.1 |
| mEBP | 19.5 | 24.3 | 19.5 | 24.9 | 18.5 | 19.3 | 17.1 | 17.5 |

*Table 3.6*: Middlebury Rankings for Occluded and Discontinuous Pixels

| | | Tsukuba | | | | Venus | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | AD | | BT | | AD | | BT | |
| | | Occ | Discon | Occ | Discon | Occ | Discon | Occ | Discon |
| Rank | EBP | 68 | 72 | 68 | 72 | 62 | 74 | 61 | 70 |
| | mEBP | 68 | 74 | 67 | 71 | 66 | 71 | 71 | 76 |

*Table 3.7*: Middlebury Rankings for Occluded and Discontinuous Pixels

| | | Teddy | | | | Cones | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | AD | | BT | | AD | | BT | |
| | | Occ | Discon | Occ | Discon | Occ | Discon | Occ | Discon |
| Rank | EBP | 79 | 82 | 79 | 82 | 78 | 76 | 77 | 80 |
| | mEBP | 76 | 75 | 75 | 75 | 82 | 76 | 77 | 75 |

As with the non-occluded regions shown earlier, the occlusion and discontinuity results reiterate the point that regardless of which method produces the best results for a particular

image, the difference is typically small. Though in both cases, the occlusion portions are more likely to be worse, but it is necessary to show these results as they provide a complete picture of the comparison being made for the above image results in Figures 3.2-3.5. In most of the cases shown in Tables 3.2- 3.7, the results show that this new modified version produces lower error percentages and higher rankings, with the majority of the improved scores being found from the results of the modified EBP using BT.

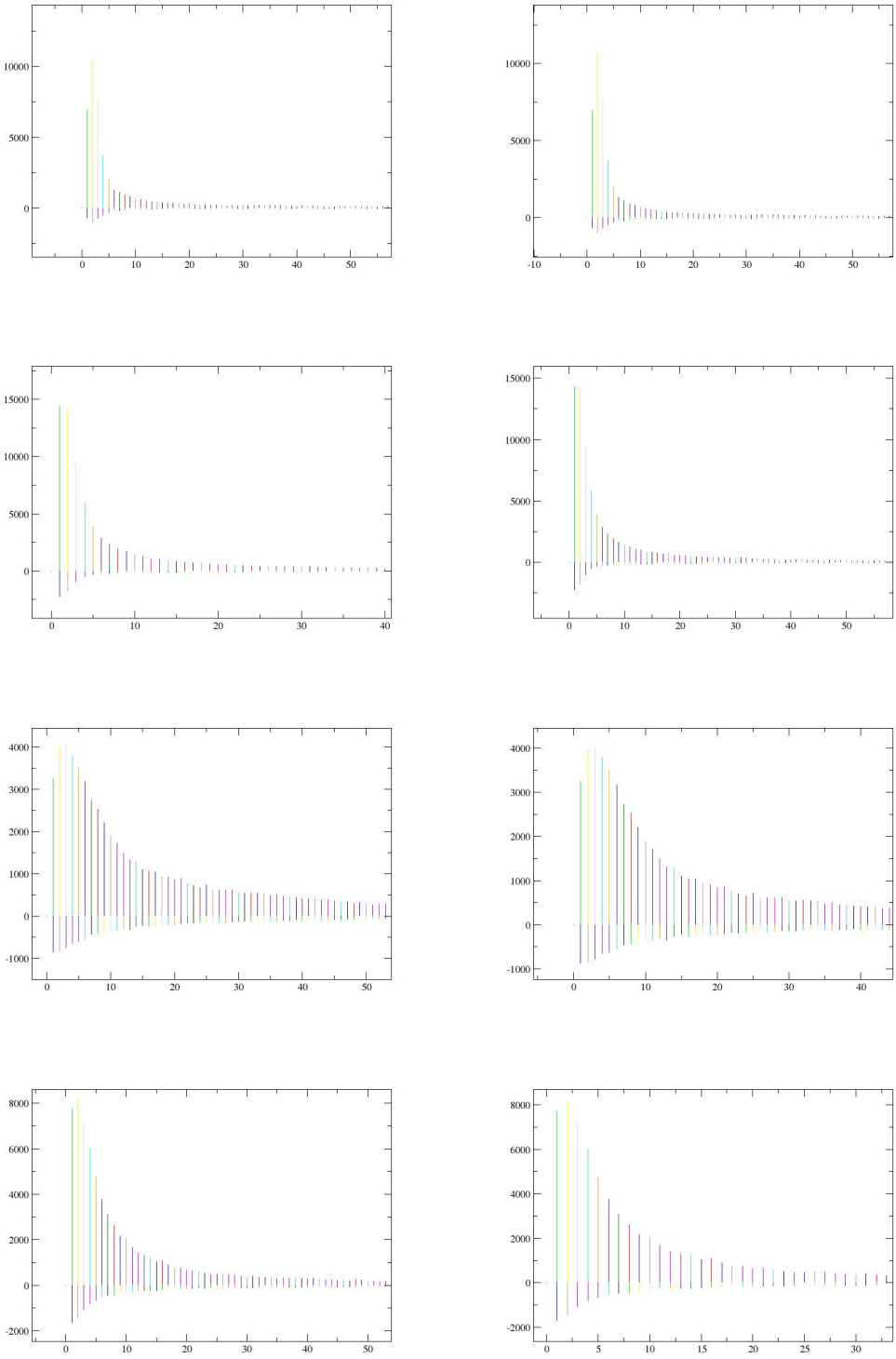



*Figure 3.6*: Histograms comparing type matches between EBP images with different data cost methods: Absolute difference (left) vs. Birchfield-Tomasi (right).

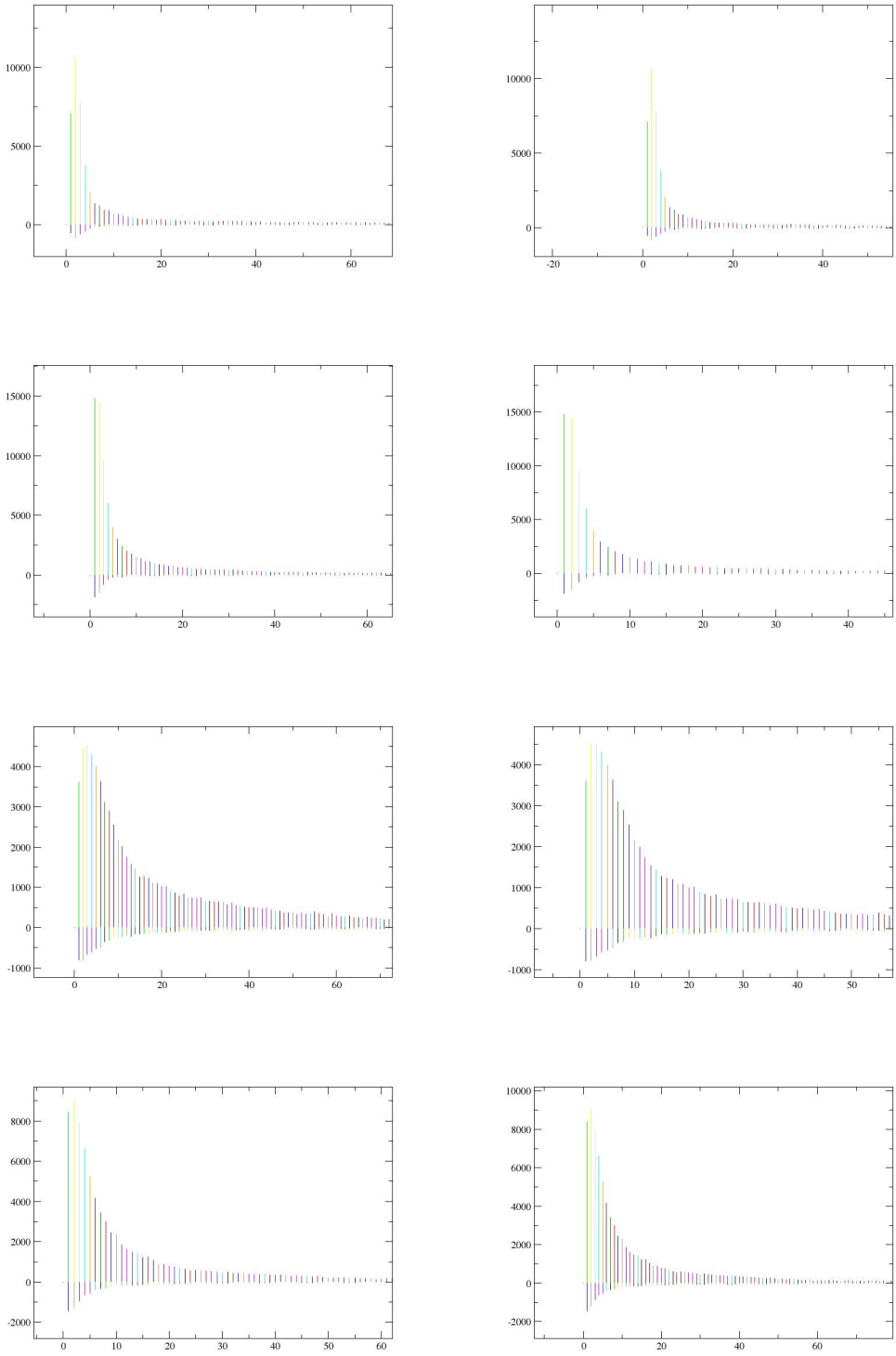


*Figure 3.7*: Histograms comparing type matches between the modified EBP images with different data cost methods: Absolute difference (left) vs. Birchfield-Tomasi (right).

In Figures 3.6 and 3.7, these histograms show the number of correct versus incorrect type matches, based on the strength of each pixel. In this case, only the extremal pixels were used to illustrate the differences between standard EBP and mEBP, as they are the group most affected by the modified efficient belief propagation. Additionally, these figures also show the differing results using absolute difference and Birchfield-Tomasi's pixel dissimilarity for calculating the data cost. In each of the histograms, there are two sets of results: the bottom bar graph showing the number of incorrect type matches and the top bar graph showing the number of correct type matches. As shown in each histogram example, the number of incorrect matches decreases as the strength of the pixels increase. This is due to several factors including the increasing gradient difference between neighboring pixels as the strength increase. This large difference affects the data cost, often resulting in larger values for the neighbors of the correct type match correspondence point; this is especially true for extremal pixels. Additionally, in the case of mEBP results, the stronger pixels have greater accuracy because two of the three main contributions in this approach were geared towards improving the stronger extremal correspondences, while indirectly influencing the weaker pixels through the propagation process.

*Table 3.8*: Accuracy of Data Type Matching for Extremal Pixels

| | Tsukuba | | Venus | | Cones | | Teddy | |
|---|---|---|---|---|---|---|---|---|
| | AD | BT | AD | BT | AD | BT | AD | BT |
| EBP | 89.0451 | 88.9125 | 88.5318 | 87.8916 | 82.7005 | 81.6843 | 84.3715 | 83.5795 |
| mEBP | 92.0712 | 91.8915 | 91.5711 | 90.9534 | 89.0033 | 88.081 | 89.7458 | 89.2968 |

Table 3.8 shows the accuracy percentages of the histograms for Efficient Belief Propagation (EBP) and the modified Efficient Belief Propagation (mEBP) using absolute difference and Birchifeld-Tomasi [2] data cost. From this table, one can see that the modified version, with its fractional disparities, performed in a slightly more accurate fashion across the board. This is due, in part, to the LevelSet-based operations, especially those that encouraged mutual correspondence between extremal pixels of the same type.

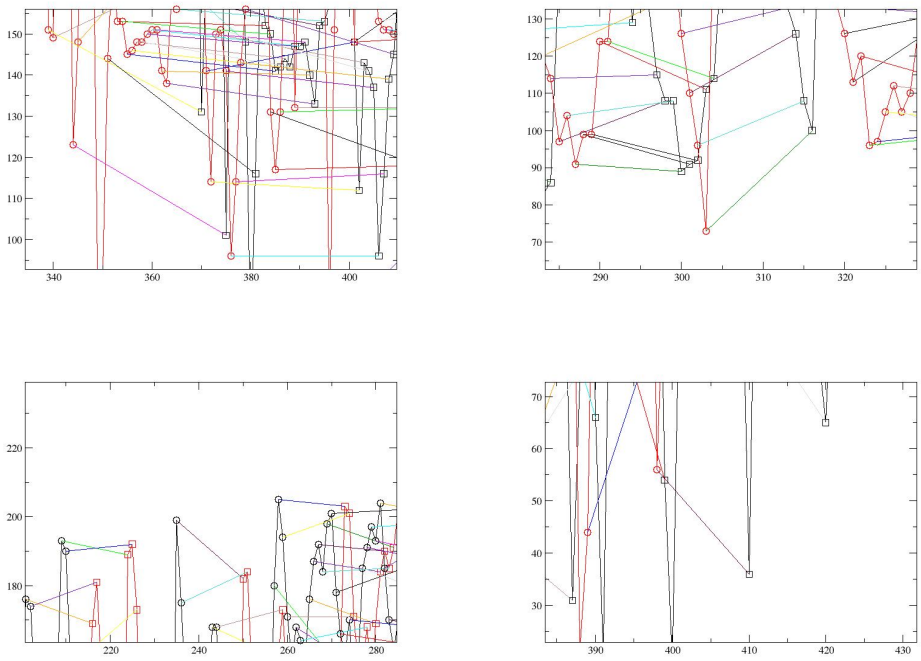



*Figure 3.8*: Profiles showing correct match examples for large intensity difference correspondence. Top row: EBP (Teddy and Venus) and bottom row: mEBP (Teddy and Venus).

Recalling the intensity difference images from Chapter 1, the large intensity difference matches within the objects' interior were suspected of being incorrect. This is, of course, not always the case, as shown in Figure 3.8. Each of these illustrations represents correspondence produced by efficient belief propagation and modified efficient belief propagation that show correct correspondence with large intensity differences. This is due to a lighting change between the cameras responsible for these images, resulting in a shift in the intensity values between many correctly corresponding points in each image.

# Chapter 4

# CONCLUSION AND DISCUSSION OF FUTURE WORK

Through the incorporation of levelset and levelset-based operations, this new approach provides a solution that not only increases the accuracy of the standard Efficient Belief Propagation, but also does not cause a significant loss in efficiency. The contributions in the new approach provide an alternative solution for stereo matching, without requiring extra, computationally expensive operations like images segmentation, plane fitting, or window-based data cost calculation. On the other hand, even with the improved accuracy provided by this new approach, there is always room for further advancement.

As for the future development in this area, there are a variety of possibilities. For instance, the development of a less computationally expensive alternative to mutual correspondence for occlusion detection is definitely worth exploring and implementing. In terms of this algorithm, it is always possible to adopt concepts, like those from Sarkis and Diepold [32], Nister et al. [49, 50], and Trinh [44], in order to improve efficiency by reducing the number of participating pixels. It may also be possible to improve the efficiency by dividing the workload using parallel processing operations through shared memory, especially for handling message computation. Alternatively, it could be possible to develop a local-based approach, incorporating principles from global approaches like belief propagation with the inherent efficiency found in local operations, to provide a faster and more accurate solution that could eventually be performed in realtime. In any case, a definite step in the future is to move from stereo matching to optical flow, because this is the problem of current interest in many application domains. In whatever direction this work proceeds, the current results show good potential for the future.

# BIBLIOGRAPHY

[1] S. Birchfield and C. Tomasi. Depth discontinuities by pixel to pixel stereo. *ICCV*, 1998.

[2] Stan Birchfield and Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4), 1998.

[3] Michael Bleyer and Margrit Gelautz. A layered stereo algorithm using image segmentation and global vsibility constraints. *ICIP*, 2004.

[4] Michael Bleyer and Margrit Gelautz. Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions. *Image Communication*, 22(2), 2007.

[5] Michael Bleyer, Margrit Gelautz, Carsten Rother, and Christoph Rhemann. A stereo approach that handles the matting problem via image warping. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11), 2001.

[7] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4), 2002.

[8] F. Crow. Summed-area tables for texture mapping. *Computer Graphics*, 18(3), 1984.

[9] Geoffrey Egnal and Richard P. Wildes. Detecting binocular half-occlusions: Empirical comparisons of five approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8), 2002.

[10] Bjorn Engquist and Stanley Osher. Stable and entropy-satifying approximations for transonic flow calculations. *Mathetmatics of Computation*, 34(149), 1980.

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, I:261–268, 2004.

[12] Pedro Felzenszwalb and David Huttenlocher. Distance transforms of sampled functions. *Cornell Computing and Information Science*, 2004.

[13] Pedro Felzenszwalb and David Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.

[14] H. Hirschmuller. Stereo vision in structured environments by consistent semi-global matching. *PAMI*, 30(2), 2008.

[15] H. Hirschmuller and D. Scharstein. Evaluation of cost functions for stereo matching. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[16] L. Hong and G. Chen. Segment-based stereo matching using graph cuts. *CVPR*, 2004.

[17] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. *ICPR*, 2006.

[18] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. *ICCV*, 2001.

[19] Chi-Hua Lai, Kun-Yuan Hsieh, Shang-Hon Lai, and Jenq Kuen Lee. Parallelization of belief propagation method on embedded multicore processors for stereo vision. *ESTImedia*, 2008.

[20] Cheon Lee and Yo-Sung Ho. Disparity estimation using belief propagation for view interpolation. *ITC-CSCC*, 2008.

[21] S. Mattoccia. A locally global approach to stereo correspondence. *3DIM*, 2009.

[22] S. Mattoccia. Accurate dense stereo by constraining local consistency on superpixels. *ICPR*, 2010.

[23] S. Mattoccia. Fast locally consistent dense stereo on multicore. *ECVW*, 2010.

[24] S. Mattoccia. Improving the accuracy of fast dense stereo correspondence algorithms by enforcing local consistency of disparity fields. *3DPVT*, 2010.

[25] S. Mattoccia. Stereo vision: Algorithms and applications (presentation). *DEIS*, 2010.

[26] S. Mattoccia, S. Giardino, and A. Gambini. Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering. *ACCV*, 2009.

[27] S. Mattoccia, F. Tombari, and L. Di Stefano. Stereo vision enabling precise border localization within a scanline optimization framework. *ACCV*, 2007.

[28] Julian McAuley and Tiberio Caestano. Exploiting data-independence for fast belief-propagation. *27th International Conference on Machine Learning*, 2010.

[29] Beddhu Murali. Stereo matching using min max sad. 2005.

[30] Alexandre Noma, Luiz Velho, and Roberto Cesar Jr. A computer-assisted colorization approach based on efficient belief propagation and graph matching. *CIARP*, 2009.

[31] Minwoo Park, Yanxi Liu, and Robert Collins. Efficient mean shift belief propagation for vision tracking. *CVPR*, 2008.

[32] Michel Sarkis and Klaus Diepold. Sparse stereo matching using belief propagation. *ICIP*, 2008.

[33] D. Scharstein and C. Pal. Learning conditional random fields for stereo. *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

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

[35] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1, 2003.

[36] Daniel Scharstein and Richard Szeliski. Middlebury data set, 2007.

[37] James A. Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, Cambridge, U.K.; New York, New York, 1993.

[38] ChangMing Sun. A fast stereo matching method. *Digital Image Computing: Techniques and Applications*, pages 95–100, 1997.

[39] J. Sun, N. Zheng, and H. Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800, 2003.

[40] Y. Taguchi, B. Wilburn, and L. Zitnick. Stereo reconstruction with mixed pixels using adaptive over-segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

[41] Marshall F. Tappen and William T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. *ICCV*, 2003.

[42] F. Tombari, S. Mattoccia, and L. Di Stefano. Segmentation-based adaptive support for accurate stereo correspondence. *PSIVT*, 2007.

[43] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda. Near real-time stereo based on effective cost aggregation. *ICPR*, 2008.

[44] Hoang Trinh. Efficient stereo algorithm using multiscale belief propagation on segmented images. *BMVC*, 2008.

[45] P. Viola and M. Jones. Robust real-time object detection. *SECOND INTERNATIONAL WORKSHOP ON STATISTICAL AND COMPUTATIONAL THEORIES OF VISION - MODELING, LEARNING, COMPUTING, AND SAMPLING*, 2001.

[46] Zeng-Fu Wang and Zhi-Gang Zheng. A region based stereo matching algorithm using cooperative optimization. *CVPR*, 2008.

[47] J. Xiao and M. Shah. Motion layer extraction in the presence of occlusion using graph cuts. *PAMI*, 27(10), 2005.

[48] Li Xu and Jiaya Jia. Stereo matching:an outlier confidence approach. *ECCV*, 2008.

[49] Qingxiong Yang, Liang Wang, Ruigang Yang, Henrik Stewenius, and David Nister. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. *IEEE Pattern Analysis and Machine Intelligence*, 2009.

[50] Qingxiong Yang, Liang Wang, Ruigang Yang, Shengnan Wang, Miao Liao, and David Nister. Real-time global stereo matching using hierarchical belief propagation. *BMVC*, 2006.

[51] Kuk-Jin Yoon and In So Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4), 2006.