Window-based, discontinuity preserving stereo

Motilal Agrawal*
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
agrawal@ai.sri.com

Larry S. Davis
Univ. of Maryland
Dept. of Computer Science
College Park, MD 20742
lsd@cs.umd.edu

Abstract

Traditionally, the problem of stereo matching has been addressed either by a local window-based approach or a dense pixel-based approach using global optimization. In this paper, we present an algorithm which combines window-based local matching into a global optimization framework. Our local matching algorithm assumes that local windows can have at most two disparities. Under this assumption, the local matching can be performed very efficiently using graph cuts. The global matching is formulated as minimization of an energy term that takes into account the matching constraints induced by the local stereo algorithm. Fast, approximate minimization of this energy is achieved through graph cuts. The key feature of our algorithm is that it preserves discontinuities both during the local as well as global matching phase.

1. Introduction

Stereo is a classical problem in computer vision with wide ranging applications. In stereo, we are given two images, I_l and I_r of a scene, S, viewed from two known positions. The goal, then, is to compute a disparity function $d(x, y)$ over the entire image $I_l(x, y)$. Although there is a volume of literature on binocular stereo with a number of algorithms that work well on many types of images, still it is considered to be a difficult problem due to several factors. The first factor is the complex interaction between light and real world surfaces, which can be specular, can have inter-reflections or be transparent. In addition, the exact position and type of light sources is usually unknown. In classical stereo, this problem is often simplified by assuming Lambertian surfaces. The second problem arises from the presence of regions of constant albedo (color) in the scene. This results in the existence of ambiguities in the disparity assignment. In order to overcome this ambiguity, assumptions about the smoothness of the disparity function are typically made. Real world scenes, however, are only "piecewise smooth" and this is the third factor which makes the problem hard. The presence of discontinuities causes occlusion and makes disparity assignment very difficult at object boundaries.

1.1. Previous work

Stereo algorithms that produce sparse disparity maps [1, 2, 21] rely on features such as edges or corners for matching. Many applications of stereo, however, require a dense disparity map. A dense disparity map may be obtained from this sparse map through interpolation. This phase is frought with difficulties and requires making some assumptions about the scene geometry. More important are stereo algorithms that produce dense disparity maps directly without the need for interpolation. The work of Zhang and Shan [27] represent an intermediate approach between the two. Theirs is an iterative matching scheme which begins with few reliable features and progressively adds matches during each iteration using the matched features as constraints. The work presented in [14, 24] follows the above philosophy of iteratively selecting features for matching with the difference that the most confident pixels are committed to at each iteration.

Stereo algorithms that produce dense disparity maps can be further classified as local or global based on the type of optimization method used. In local methods, the disparity value at each pixel is chosen independently of other pixels. Since the raw error function of assigning a disparity to a pixel is noisy, the error function is usually aggregated over a local window. The simplest technique is to use square windows of fixed size [17]. Such algorithms often assume that all the pixels within this window have the same disparity. The work of Okutomi and Kanade [13] makes use of adaptive windows. Here, the window size is varied and at each

* This work was carried out while the author was a graduate student at the University of Maryland

pixel the size of the window is chosen so as to minimize the disparity uncertainty at that pixel. Geiger et al [8] and Fusiello et al [7] use a multiple window method where a limited number of distinct windows are tried for each pixel and disparity, and the one with best correlation is retained. This is also the idea behind spatially shiftable windows [5, 14, 25]. The work of Vekseler [26] proposes to overcome the shape restriction of square/rectangular windows by optimizing matching cost over a large class of "compact" windows.

Since these local aggregation methods assume a constant disparity in these windows, they do not perform well in regions of depth discontinuities. The different approaches of multiple windows, spatially shiftable windows and "compact windows" is an attempt to overcome this problem by varying the shape and size of these windows. Global optimizations methods attempt to overcome this problem by minimizing a certain energy function. The energy function is a combination of a "data term" and a "smoothness term". Several different methods for the minimization of these energy functions have been used including simulated annealing [9], relaxation labelling [23] and non-linear diffusion of support [19]. Over the last few years, several algorithms for energy minimization based on graph cuts have been presented. When there are only two disparity labels, Greg et al [10] showed how to find the global minimum using a single graph cut. For the multi-disparity case [18, 11, 12, 4] used graph cuts to find the exact global optimum of certain types of energy functions. Their energy function, however, is not discontinuity preserving. Boykov et al [6] have presented approximate algorithm with a guarantee on bounds for discontinuity preserving energy functions. This has been generalized to enforce the uniqueness constraint by Kolomogorov et al [15].

In this paper, we combine local window-based methods into a global optimization framework using graph cuts. The simplest way of doing this, as proposed in [14], is to use the aggregate error term obtained from window-based local stereo as the "data term" in the global energy function. However this ignores the dependencies that exist within this window. The assignment of a particular disparity $d(x, y) = d_0$ at a pixel (x, y) constrains the disparities at all other pixels within the window centered at (x, y). For example, if we used the simple aggregate term with constant disparity then all pixels within the window should also have disparity d_0. This will however lead to smoothing across discontinuities, unless the local assignment of disparities takes account of this. Our window-based local stereo algorithm, which is presented in Section 2, avoids this smoothing across discontinuities. The global minimization framework, which is discussed in Section 3, then uses this local stereo to produce a global labelling which is maximally consistent with the labelling induced by local stereo. This can be interpreted as minimizing a certain energy function. This energy function is presented in Section 3.1. Approximation algorithm for minimization of this energy function using graph cuts is next presented in Section 3.3. Results on real data are presented in Section 4. Finally, Section 5 concludes the paper with suggestions for future work.

2. Local stereo

2.1. Preliminaries

I_l and I_r are the left and right images of the stereo pair. It is assumed that the input images are rectified. The disparity at pixel (x, y) is denoted by $d(x, y)$. The disparity function can take one of the K integer values between the disparity limits of the scene. At the base of any stereo algorithm is an error function which denotes the error of assigning a disparity α to a pixel (x, y). This will be denoted by $C(x, y, \alpha)$. The simplest such error function uses the absolute difference of the pixel intensities in the left and right images i.e. $C(x, y, \alpha) = |I_l(x, y) - I_r(x - \alpha, y)|$. Other error functions involve interpolating the intensities to avoid sampling artifacts [3]. $W(x, y)$ denotes a window W centered at pixel (x, y). W_h is a square window with dimensions $(2h + 1) \times (2h + 1)$. In other words, $W_h(x, y) = \{(i, j) : |i - x| \leq h \quad \text{or} \quad |j - y| \leq h\}$.

2.2. Bi-labelled windows

In traditional window-based local stereo, windows centered at each pixel are considered. It is assumed that all the pixels in that window have a constant disparity. The error term is aggregated over the window to give the matching cost for assigning that disparity to the center of the window. The disparity which minimizes this matching cost is then assigned to the center. In mathematical terms,

$$d(x, y) = \arg\min_{\alpha} C_W(x, y, \alpha) \quad (1)$$

$$C_W(x, y, \alpha) = \sum_{(i,j) \in W(x,y)} C(i, j, \alpha) \quad (2)$$

The assumption of constant disparity within the window is a fundamental limitation of such approaches, which results in an overly smooth disparity map, particularly at object boundaries. Local windows, of course, do not have constant disparity; they generally have a few number of disparity levels. In other words, the range of disparities present within local windows is small. Table 1 illustrates this point using the University of Tsukuba ground truth disparities. At each pixel the total number of disparities present within the window centered on that pixel was counted for windows of different sizes. The window sizes are listed horizontally and the number of disparities present in the window is listed vertically. Each entry in the table corresponds to the percentage

Disparity count	Window size	3×3	5×5	7×7	9×9	11×11	13×13	21×21
1		95.21	86.45	79.05	72.82	67.30	62.41	47.58
2		4.74	13.05	19.48	24.41	28.40	31.49	37.35
3		0.05	0.49	1.40	2.59	3.93	5.42	11.89
4		0.00	0.01	0.07	0.18	0.37	0.68	3.14

Table 1. Disparity variation within local windows

of pixels which have a particular number of disparities for that window size. For example, the table shows that for window of size 7×7, only 1.40% of the pixels will have 3 disparities for the pixels within that window. Our local stereo algorithm exploits the limited number of disparities present in small windows. We assume that within these windows, there are at most two disparities present. The motivation behind the assumption of two disparities is two-fold. Firstly, this corresponds intuitively to the idea of a background and foreground disparity. Additionally, when there are only two disparity labels, the *global* minimum of the corresponding discontinuity preserving energy function can be found very efficiently using a single graph cut [10]. Intuitively, the assumption that the local windows are bi-labelled can be seen as a discontinuity preserving smoothness constraint and a generalization of the assumption that all the pixels within a local window have a single disparity label. In the next subsection, we give a sketch of how to perform this minimization for the case of two labels. Details can be found in [10, 11, 6].

2.3. Graph cut for exact minimization of bi-labelled disparities

In the case of two labels, (α, β), the labelling corresponding to minimum energy is found by finding a minimum cut through a certain graph. Each pixel (x, y) corresponds to a node p of the graph. In addition, there are two additional nodes corresponding to the source (S_0) and sink (S_1). There are edges from the source to every node p with weight $S_0 p = C(x, y, \alpha)$ and also edges from p to the sink with edge weights $pS_1 = C(x, y, \beta)$. In addition, for every pair of neighbors (x, y) and $(x-1, y)$ or $(x, y-1)$ there is an edge connecting the corresponding nodes p and q with edge weight $U_l(p, q)$, where $U_l(p, q)$ is a penalty for assigning different labels to neighboring nodes (p, q). A minimum cut of this graph then corresponds to a labelling with minimum energy.

2.4. Local stereo using bi-labelled windows

The above local stereo for two labels is applied to windows centered on each pixel (x, y) for each disparity pair (α, β), $\alpha \neq \beta$. For every such pair of disparities, an assignment $d^{\alpha\beta}_{(x,y)}(i, j) \quad \forall (i, j) \in W(x, y)$ is obtained. The aggregate cost of the local disparity function ($d^{\alpha\beta}_{(x,y)}$) is denoted by $C^{\alpha\beta}_W(x, y)$ and is calculated as $C^{\alpha\beta}_W(x, y) = \sum_{(i,j)} C\left(i, j, d^{\alpha\beta}_{(x,y)}(i, j)\right)$ where $(i, j) \in W(x, y)$. This is simply the sum of the costs of assigning the disparity $d^{\alpha\beta}_{(x,y)}(i, j)$ over all the pixels in the window centered at (x, y). Let $d^{\alpha}_{(x,y)}$ denote the minimum cost disparity assignment in the window centered at (x, y) such that the center (x, y) gets a label α i.e. $d^{\alpha}_{(x,y)}(x, y) = \alpha$ and let $C^{\alpha}_W(x, y)$ be the corresponding minimum aggregate cost function. Then, $d^{\alpha}_{(x,y)}$ can be computed by finding among all β, the one that gives minimum $C^{\alpha\beta}_W(x, y)$, provided, of course, $d^{\alpha\beta}_{(x,y)}$ assigns a label α to (x, y). Mathematically,

$$\gamma = \arg\min_{\beta} C^{\alpha\beta}_W(x, y) \text{ and } d^{\alpha\beta}_{(x,y)}(x, y) = \alpha \tag{3}$$

$$d^{\alpha}_{(x,y)} = d^{\alpha\gamma}_{(x,y)} \text{ and } C^{\alpha}_W(x, y) = C^{\alpha\gamma}_W(x, y) \tag{4}$$

Note that it may happen that there does not exist any β such that $d^{\alpha\beta}_{(x,y)}(x, y) = \alpha$. In that case (x, y) will never get a label α and $C^{\alpha}_W(x, y) = \infty$.

2.5. Computational considerations

Computing and storing $d^{\alpha}_{(x,y)}$ for each pixel (x, y) and α can be computationally as well as memory intensive. To make this computation tractable, we instead compute the labelling of disparities over the entire image for each disparity pair $(\alpha\beta)$ to produce a disparity assignment over the entire image ($d^{\alpha\beta}_g$). The bi-labelled disparity assignment for a particular pixel $d^{\alpha\beta}(x, y)$ and the aggregated cost $C^{\alpha\beta}_W(x, y)$ may then be extracted by considering the assignments of labels in the window $W(x, y)$ centered at (x, y). i.e. $d^{\alpha\beta}_{(x,y)}(i, j) = d^{\alpha\beta}_g(i, j) \quad \forall (i, j) \in W(x, y)$. As already pointed out, minimization using two labels is exact, irrespective of the size of the window. Thus, the only drawback of this scheme is that the pixels which are on the boundary of the window influence the labelling inside the window. As the size of the window increases, the proportion of boundary pixels decreases and thus this boundary influence

decreases. For a sufficiently large window size, this "boundary" effect will only be significant in untextured windows, wherein, the disparity labelling is ambiguous.

3. Global minimization framework for windows

The local window based stereo algorithm computes for each pixel (x, y) and each disparity α, the aggregated total cost $C^{\alpha}_{W}(x, y)$ and the local disparity assignments $d^{\alpha}_{(x,y)}$. The goal of the global stereo algorithm is to assign disparities to each pixel, $d(x, y)$, in a manner consistent with the local disparity assignments. That is, if a pixel (x, y) is assigned a disparity α, then all the pixels in $W(x, y)$ must have disparities defined by $d^{\alpha}_{(x,y)}$. We accomplish this also using a graph cut algorithm. Two features of our algorithm are 1. the corresponding discontinuity energy term is based on the pott's energy of assignments [15]. 2. the neighborhood relation of nodes in the graph is defined to be across the entire window rather than adjacent pixels. It is also worth emphasizing the fact that this algorithm is independent of the particular local algorithm used to compute $d^{\alpha}_{(x,y)}$ as long as the local algorithm is discontinuity preserving.

3.1. Energy function

The energy of a labelling d can be calculated as the sum of the energies of the individual pixels : $E(d) = \sum_p E_l(\alpha)$ where p denotes the single pixel (x, y) and $\alpha = d(p)$ is the label of p under d. The energy E_l is composed of two terms: the data term E_{dat} and the penalty term E_{pen}. For the data term, the aggregated local cost $C^{\alpha}_{W}(p)$ is the natural choice. The trick, however, is in the proper choice of the penalty term so as to make the minimization tractable. Consider all the pixels q in the neighborhood $W(p)$ of p. The disparity at q, as determined by the local disparity assignment in the window centered at p is $d^{\alpha}_{p}(q)$ and $d(q)$ is the disparity at q under the labelling d. These two disparities must conform with each other. Therefore, there is a positive penalty incurred if $d(q) \neq d^{\alpha}_{p}(q)$. Let this penalty be denoted by $U_g(p, q)$. One choice of the penalty term would then be

$$E^{1}_{\text{pen}}(p) = \sum_{q \in W(p), d^{\alpha}_{p}(q) \neq d(q)} U_g\,(p, q) \tag{5}$$

The same constraint can also be expressed through a different penalty function

$$E^{2}_{\text{pen}}(p) = \sum_{q \in W(p)} U_g\,(p, q)\,.T\left(d(q) \neq d^{\alpha}_{p}(q)\right) \tag{6}$$

where $T(.)$ is 1 if its argument is true and 0 otherwise. Basically, the above function imposes a penalty $U_g\,(p, q)$ if the assignment $d^{\alpha}_{p}(q)$ is not present in the current labelling. This energy function is different from the standard Potts discontinuity energy and is similar to Potts energy on assignments used in [15]. Thus the energy of a labelling d is

$$E(d) \;=\; \sum_{p} E_{\text{dat}}(p) + \sum_{q \in W(p)} U_g\,(p, q)\,.T\left(d(q) \neq d^{\alpha}_{p}(q)\right) \tag{7}$$

Exact minimization of this energy function can be shown to be NP-hard. Therefore, we need to consider approximate algorithms for this minimization. In the next section, we give details about how alpha expansion [6] moves can be used to perform this minimization efficiently.

3.2. Minimization using expansion moves

The single step of the expansion move algorithm is called α-expansion. Suppose that we have some current labelling d and we are considering a label α. The α-expansion move results in a new labelling d' and satisfies the property that for any pixel p either $d'(p) = d(p)$ or $d'(p) = \alpha$. That is, the pixel may retain its old label or change its label to α. The label α is repeatedly chosen in some order (fixed or random) and the α-expansion move is then applied. If this move results in a decrease in the total energy, then we simply change our current labelling to d'. If there is no such α which results in a decrease of energy, we are done. The critical step in this algorithm is to efficiently compute the α-expansion move resulting in the largest reduction in the energy. Kolomogorov et al [16] have characterized the class of energy functions that can be minimized by graph cuts. For the problem of pixel labelling using α-expansion moves, the energy function is required to be a metric. For the case of metric energy functions (eg. Potts energy model), other authors have used graph cuts to find the *largest* reduction in the energy [15, 6]. It can be shown that our energy function (equation 7) is not metric and hence a graph cut based approach is not guaranteed to find the *largest* decrease in energy. Nevertheless, graph cuts can still be used to find *large* reductions in the energy and in the next section, we show how α-expansion moves can be used to minimize this energy.

3.3. α-expansion using graph cuts

For a current labelling d and a disparity α, a directed graph $\mathcal{G}_{\alpha} = \langle \mathcal{V}_{\alpha}, \mathcal{E}_{\alpha} \rangle$ is constructed. Each pixel corresponds to a node of the graph. In addition there are two special nodes α and $\bar{\alpha}$ which are the source and the sink nodes for the maximum flow computation. There is an edge from the source to each node of the graph t^{α}_{p} and the edge $t^{\bar{\alpha}}_{p}$ connects p to the sink. For every pair of pixels p and q such that $p \in W(q)$ (or equivalently $q \in W(p)$) there

edge	weight	for
$t_p^{\bar{\alpha}}$	∞	$d(p) = \alpha$
$t_p^{\bar{\alpha}}$	$C_W^{\beta}(p) + \sum_{q \in W(p)} U_g(p,q)$	$d(p) = \beta, \beta \neq \alpha$ $d_p^{\beta}(q) \neq \alpha$ and $d_p^{\beta}(q) \neq d(q)$
t_p^{α}	$C_W^{\alpha}(p) + \sum_{q \in W(p)} U_g(p,q)$	$d_p^{\alpha}(q) \neq \alpha$ and $d_p^{\alpha}(q) \neq d(q)$
$e_{\{p,q\}}$	$U_g(p,q)\,(T(X) + T(Y))$	$X \equiv d_q^{\alpha}(p) = \alpha$ $Y \equiv d_p^{\beta}(q) = d(q), \beta = d(p)$
$e_{\{q,p\}}$	$U_g(q,p)\,(T(X) + T(Y))$	$X \equiv d_p^{\alpha}(q) = \alpha$ $Y \equiv d_q^{\beta}(p) = d(p), \beta = d(q)$

Table 2. Edge weights for α-expansion

are directed edges $e_{\{p,q\}}$ and $e_{\{q,p\}}$. The weights of these edges are given in Table 2. These particular weights correspond to the energy function in equation 7. It is easy to see that the minimum cut $\mathcal{C}$ on $\mathcal{G}_\alpha$ is then one α-expansion away from d. The new labelling corresponding to this cut then is d', where $d'(p) = \alpha$ if $t_p^{\alpha} \in \mathcal{C}$ and $d'(p) = d(p)$ if $t_p^{\bar{\alpha}} \in \mathcal{C}$. As pointed out earlier, since our energy function is not a metric, the graph cut is not guaranteed to find the α-expansion move resulting in the largest decrease in energy. However, in practice the performance of this greedy minimization technique is quite good and is presented in the next section.

4. Results

There has been a need in the computer vision community to set up a test bed for quantitative evaluation and comparison of different stereo algorithms. Towards this end, Scharstein and Szeliski [20] have set up test data along with ground truth which is available at their website (`www.middlebury.edu/stereo`). We have evaluated our proposed algorithm on these test data sets. The metric used for evaluating the algorithm is the percentage of bad matching pixels

$$B = \frac{1}{N} \sum_{(x,y)} \left(|d_C(x,y) - d_T(x,y)| > 1 \right) \qquad (8)$$

Here $d_T(x,y)$ is the true disparity at pixel (x,y) and $d_C(x,y)$ is the calculated disparity by the proposed algorithm. This measure B is calculated at various regions of the input image which have been classified as untextured (untex), discontinuity (disc) and the entire image (all).

For our data term (error in assigning a disparity to a pixel), we have used the technique of Birchfield and Tomasi [3, 6, 20] to obtain an error term that is insensitive to image sampling. This is accomplished by taking the minimum of the pixel matching score and the score at $\pm\frac{1}{2}$-step displacements or 0 if there is a sign change in either interval. This matching score is then truncated to a maximum value to obtain a robust matching score. For color images, this measure is computed for each channel separately and the final score is the average of the scores in each channel.

Our algorithm has three parameters. The first is the size of the window to be used. In our experiments we have used windows of size 7×7 and 11×11. In addition, we have two lambda's, λ_l is the parameter that controls the level of smoothness in local stereo optimization and λ_g controls the smoothness for global optimization. Following other authors [6, 20, 15], we have used static cues in order to align the discontinuities along edges. This is achieved by using the following function for the penalty term

$$U(p,q) = \begin{cases} 2\lambda & : \text{ if } |I_p - I_q| \leq 5 \\ \lambda & : \text{ if } |I_p - I_q| > 5 \end{cases} \qquad (9)$$

This penalty term is used both in the local (U_l) and global (U_g) optimization steps with the appropriate lambda's (λ_l or λ_g).

For all our experiments $\lambda_l = 8$. For 7×7 windows, best results were obtained using $\lambda_g = 1.0$ and for 11×11 windows, $\lambda_g = 0.8$. The cost of labelling a pixel as occluded was fixed at 20 in all cases. Table 3 shows the percentage of bad matching pixels obtained as a result of applying our algorithm to the four data sets available. For comparison purpose, the results obtained from other dense stereo algorithms have also been included. These are Belief Propagation (Belief) [22], Graph Cuts (GraphCut) [6], Graph Cuts with occlusion (GraphCutOcc) [15] and compact windows (CompWin) [26]. From the table it is clear that except for the venus dataset, the error rate is lower for window size 11 as compared to using windows of size 7. Also, for window size 11, the results are better than using a global graph cut alone (GraphCut). Overall, the results of our algorithm on these datasets was comparable to other "state of the art" stereo algorithms listed on the stereo website

Algorithm	Tsukuba			Sawtooth			Venus			Map	
	all	untex	disc	all	untex	disc	all	untex	disc	all	disc
Window 7	1.88	1.29	10.01	1.46	0.17	5.42	1.36	1.56	8.65	0.48	5.15
Window 11	1.78	1.22	9.71	1.17	0.08	5.55	1.61	2.25	9.06	0.32	3.33
Belief	1.15	0.42	6.31	0.98	0.30	4.83	1.00	0.76	9.13	0.84	5.27
GraphCut	1.94	1.09	9.49	1.30	0.06	6.34	1.79	2.61	6.91	0.31	3.88
GraphCutOcc	1.27	0.43	6.90	0.36	0.00	3.65	2.79	5.39	2.54	1.79	10.08
CompWin	3.36	3.54	12.91	1.61	0.45	7.87	1.67	2.18	13.24	0.33	3.94

Table 3. Results for stereo

(`www.middlebury.edu/stereo`). The disparity map obtained using a 11×11 window is shown in Figure 1 along with the ground truth disparity map. Typical running time for window size 11 is a few minutes on a Pentium 1.7 Ghz machine.

5. Conclusion

By observing that there are usually no more than two disparity levels in a small neighborhood of individual pixels, we have presented a graph cut based algorithm for combining window-based local stereo into a global optimization framework. For local bi-labeling, the graph cut method is applied to find the best score of all possible disparity pairs. The optimal solution is used in the global optimization to compute the compatibility of disparity values between neighboring pixels. Instead of the smoothing energy term usually used in MRF, the bi-labeling is used to support the final disparity map that agrees with the local bi-labeling results, thereby preserving the disparity discontinuities. We have applied our algorithm on several stereo data sets and have presented quantitative evaluation of our method with several others. Experiments on real data indicate that our results are comparable to other pixel-based as well as window-based stereo methods. Our approach of combining window based local stereo into a global optimization framework using graph cuts fits in well with other schemes and takes a step towards completing the matrix of stereo algorithms.

Some of the key areas of future work are

1. Computational speedup: As discussed in Section 2.5, for computational speedup, instead of computing the bi-labelling for windows around all the individual pixels, the bi-labeling need only be computed on the entire image. However, for large number of disparities, even this can be computationally prohibitive. We can usually limit the disparities that can be assigned to each pixel based on the error of assigning that disparity to the pixel. And therefore, we only have to try those pairs of disparities that can be assigned to a particular pixel. Another approach that can be applied is to find all the disparities assigned to a pixel by varying the smoothness parameter of a global algorithm and then we can choose the pairs of disparities that can be assigned to each pixel from these assignments.

2. Selection of window size: Our paper considers fixed size windows. For this scenario, the appropriate size of the window to be used plays a very important role. Automatic selection of the window size to be used for a given stereo pair is an important area of future work. In addition, it is easy to see that no fixed window size will work well for all areas of the image. In particular, our local window based stereo algorithm relies on the assumption that local windows have at most two disparities, which will fail for large sizes of window. Therefore, ideally, we would like to automatically select the window size at each pixel. Incorporation of variable window size is likely to increase the performance of our algorithm.

3. Generalizing bi-labelled local stereo: Our local window based stereo is based on the assumption that the window is bi-labelled. Although this assumption is valid for most of the image (provided the window size is small), still it is not general enough and needs to be generalized.

4. Incorporating occlusion & uniqueness constraint: Our global optimization framework does not currently handle the uniqueness constraint in the right image. Kolomogorov and Zabih [15] have incorporated this constraint using a different graph construction. It is easy to incorporate this method of graph construction in our global minimization framework. This, we believe, will lead to further improvements.

6. Acknowledgements

We are grateful to Y. Ohta and Y. Nakamura for supplying the ground truth imagery for the University of Tsukuba data. We are also grateful to D. Scharstein and R. Szeliski

[20] for putting together the stereo evaluations website. Financial support from the "Robotics CTA" under contract #9614P with General Dynamics Robotics Systems is gratefully acknowledged.

References

[1] M. Agrawal, D. Harwood, R. Duraiswami, L. Davis, and P. Luther. Three dimensional ultrastructure from transmission electron microscope tilt series. In *Proc. Indian Conference on Vision Graphics and Image Processing*, December 2000.

[2] S. Alibhai and S. W. Zucker. Contour-based correspondence for stereo. In *Proc. Sixth European Conference on Computer Vision*, volume I, pages 314–330, 2000.

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

[4] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Proc. Seventh International Conference on Computer Vision*, pages 489–495, Sept. 1999.

[5] A. Bobick and S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):1–20, September 1999.

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

[7] A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 858–863, June 1997.

[8] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. In *Proc. European Conference on Computer Vision*, pages 425–433, 1992.

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

[10] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51:271–279, 1989.

[11] H. Ishikawa. *Global Optimization Using Embedded Graphs*. PhD thesis, New York University, May 2000.

[12] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *Proc. Fifth European Conference on Computer Vision*, pages 232–248, June 1998.

[13] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, September 1994.

[14] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 103–110, December 2001.

[15] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proc. International Conference on Computer Vision*, volume II, pages 508–515, July 2001.

[16] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *Proc. Seventh European Conference on Computer Vision*, volume III, pages 65–81, May 2002.

[17] M. Okutomi and T. Kanade. A multiple-baseline stereo. In *Proc. IEEE conference on Computer Vision and Pattern Recognition*, pages 63–69, 1991.

[18] S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Proc. Sixth International Conference on Computer Vision*, pages 492–499, 1998.

[19] D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. In *Proc. IEEE conference on Computer Vision and Pattern Recognition*, pages 343–350, 1996.

[20] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 131–140, December 2001.

[21] Y. Shan and Z. Zhang. Corner guided curve matching and its application to scene reconstruction. In *Proc. IEEE conference on Computer Vision and Pattern Recognition*, pages I:796–803, 2000.

[22] J. Sun, H.-Y. Shum, and N.-N. Zheng. Stereo matching using belief propagation. In *Proc. Seventh European Conference on Computer Vision*, volume II, pages 510–524, May 2002.

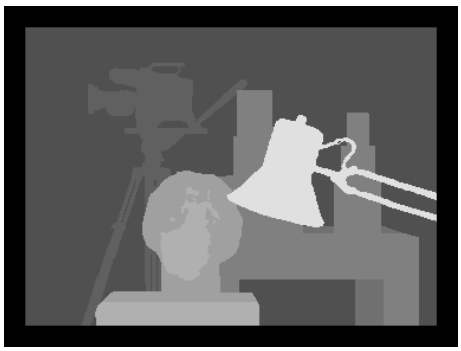
[23] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *International Journal of Computer Vision*, 5(3):271–302, December 1990.

[24] R. Szeliski and P. Golland. Stereo matching with transparency and matting. In *Proc. Sixth International Conference on Computer Vision*, pages 517–524, 1998.

[25] H. Tao, H. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *Proc. International Conference on Computer Vision*, volume I, pages 532–539, 2001.

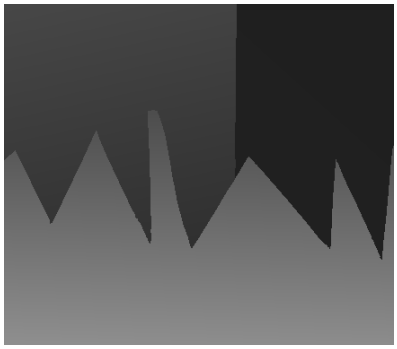
[26] O. Vekseler. Stereo matching by compact windows via minimum ratio cycle. In *Proc. International Conference on Computer Vision*, volume I, pages 540–547, 2001.

[27] Z. Zhang and Y. Shan. A progressive scheme for stereo matching. In *Springer LNCS 2018: 3D Structure from Images - SMILE 2000, M. Pollefeys et al. (eds.)*, pages 68–85. Springer-Verlag, 2001.

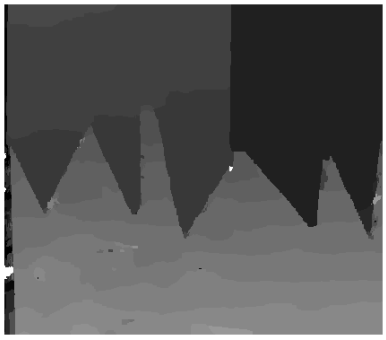


(a) Tsukuba (Ground truth)

(b) Tsukuba (Our result)

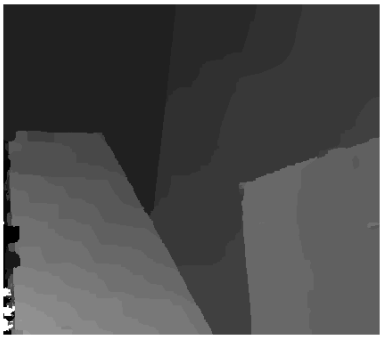


(c) Sawtooth (Ground truth)



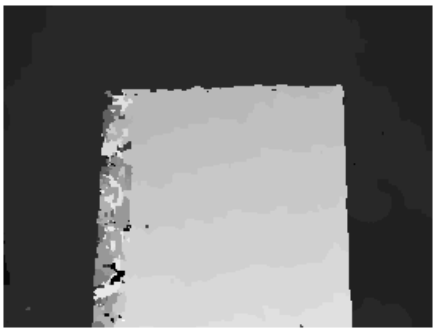
(d) Sawtooth (Our result)

(e) Venus (Ground truth)



(f) Venus (Our result)

(g) Map (Ground truth)



(h) Map (Our result)

Figure 1. The results of applying our algorithm to different data sets for a 11×11 window size.