Comparison of Graph Cuts with Belief Propagation for Stereo, using Identical MRF Parameters

Marshall F. Tappen   William T. Freeman

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
{mtappen,billf}@ai.mit.edu

Abstract

Recent stereo algorithms have achieved impressive results by modelling the disparity image as a Markov Random Field (MRF). An important component of an MRF-based approach is the inference algorithm used to find the most likely setting of each node in the MRF. Algorithms have been proposed which use Graph Cuts or Belief Propagation for inference. These stereo algorithms differ in both the inference algorithm used and the formulation of the MRF. It is unknown whether to attribute the responsibility for differences in performance to the MRF or the inference algorithm. We address this through controlled experiments by comparing the Belief Propagation algorithm and the Graph Cuts algorithm on the same MRF's, which have been created for calculating stereo disparities. We find that the labellings produced by the two algorithms are comparable. The solutions produced by Graph Cuts have a lower energy than those produced with Belief Propagation, but this does not necessarily lead to increased performance relative to the ground-truth.

1. Introduction

Two of the more exciting recent results in computational vision have been the development of fast algorithms for approximate inference in Markov Random Fields (MRF's): Graph Cuts [5] and Belief Propagation [16]. Papers on both graph cuts and belief propagation have won recent academic recognition [8, 9, 16] and have been applied to a number of problems [6, 7]. In the realm of stereo, the top contenders for the best stereo shape estimation, on the most common comparison data, either use Belief Propagation [11] or Graph Cuts [3, 5]. Both algorithms allow fast, approximate solutions to MRF's, which are powerful tools for modelling vision problems, but intractable to solve with

reasonable speed until recently. These algorithms may become the basis for new and powerful vision algorithms, so it is important to know how they compare against each other. The stereo problem provides a well-understood test-bed for comparison.

Unfortunately, the competing stereo algorithms use both a different inference algorithm and a different formulation of the MRF. This raises the question of how to understand differences in systems' performance. Labelling an MRF has been shown to be NP-hard, so both Graph Cuts and Belief Propagation approximate the optimal solution. Should one system's improvement over the other be attributed to its choice of an inference algorithm? Alternatively, does most of the improvement belong to the authors' unique formulation of the MRF?

The answer to these questions is important because advancing the field of computer vision and building on these two systems requires understanding what makes these algorithms different and how these differences affect the systems' performance. To answer this question, we show a controlled comparison of the Belief Propagation and Graph Cuts algorithms. The two algorithms are examined on identical MRF's, allowing us to measure the quality of the solutions produced by the two algorithms and isolate the effects of the inference algorithms on system performance.

In Section 2 we discuss how the MRF model can be used to calculate stereo disparities. Section 3 explains the formulation of the MRF's used in our tests and the implementation of the Belief Propagation and Graph Cuts algorithms. The results of our comparison are presented in Section 4 and are discussed in Section 5.

2. MRF Model for Stereo

Given a rectified stereo pair of images, the goal is to find the disparity of each pixel in the reference image. In

1

[10], Scharstein and Szeliski point out that most stereo algorithms perform four basic steps:

1. Matching cost computation

2. Cost (or support) aggregation

3. Disparity optimization

4. Disparity refinement

In this section, we discuss how steps 1, 2, and 3 can be accomplished by modelling the disparity image as a Markov Random Field.

2.1. Matching Cost Computation

The true disparity of each pixel in the disparity image is a random variable, denoted x_p for the variable at pixel location p. Each variable can take one of N discrete states, which represent the possible disparities at that point. For each possible disparity value, there is a cost associated with matching the pixel to the corresponding pixel in the other stereo image at that disparity value. Typically, this cost is based on the intensity differences between the two pixels, y_p. This cost is reflected in the compatibility function, $\Phi(x_p, y_p)$, which relates how compatible a disparity value is with the intensity differences observed in the image. Smaller intensity differences will correspond to higher compatibilities and vice-versa.

2.2. Support Aggregation

The next step is to aggregate support for the candidate disparities. A standard sum-of-squared-differences algorithm accomplishes this by assuming a constant disparity over a small window surrounding each point and finding the best matching cost [10]. A MRF approach aggregates support by introducing a second compatibility function, $\Psi(\cdot)$. This function expresses the compatibility between neighboring variables. Traditionally, only variables adjacent to a particular variable are considered its neighbors. Therefore, every $\Psi(\cdot)$ is of the form $\Psi(x_p, x_n)$, where the location n is adjacent to p. This is known as a pair-wise Markov Random Field. Typically only pairwise Markov Random Fields are used for stereo problems because considering more neighbors quickly makes inference on the field computationally intractable. Although the compatibility functions only consider adjacent variables, each variable is still able to influence every other variable in the field via these pair-wise connections.

2.3. Disparity Optimization

With the compatibility functions defined, the joint probability of the MRF can be written as [1]:

$$P(x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N) =$$
$$\prod_{(i,j)} \Psi(x_i, x_j) \prod_p \Phi(x_p, y_p) \qquad (1)$$

where N is the number of nodes, (i, j) represent a pair of neighboring nodes, x_n is the variable at location n, and y_n is the variable representing the intensity differences. The y variables are observed and therefore fixed during optimization.

The disparity optimization step requires choosing an estimator for $x_1 \ldots x_N$. The two most common estimators are the Minimum Mean Squared Error (MMSE) estimator and Maximum A Posteriori (MAP) estimator. The MMSE estimate of each x_i is the mean of the marginal distribution of x_i. The MAP estimate is the labelling of $x_1 \ldots x_N$ that maximizes Equation 1. For this comparison, we use the MAP estimator because the Graph Cuts algorithm is designed to compute the MAP estimator. In Section 5.2, we discuss the advantages of using the MMSE estimator.

2.4. Equivalence to Energy Minimization

As posed above, the best disparities are found by maximizing a probability. Taking the log of Equation 1, we see finding the MAP estimate is equivalent to minimizing a function of the form

$$E(x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N) =$$
$$\sum_{(i,j)} -\log \Psi(x_i, x_j) + \sum_p -\log \Phi(x_p, y_p) \qquad (2)$$

In [5], this equation is expressed as

$$E(x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N) =$$
$$\sum_{(i,j)} V(x_i, x_j) + \sum_p D(x_p, y_p) \qquad (3)$$

The functions $V(\cdot)$ and $D(\cdot)$ are energy functions. The fact that maximizing the probability in Equation 1 is equivalent to minimizing the energy in Equation 3 is important because it means that the Belief Propagation and the Graph Cuts algorithms are attempting to solve the same problem. Once the MRF has been formulated, one algorithm can be substituted for the other in the stereo algorithms.

3. MRF Formulation

To determine whether using one algorithm presents a clear advantage over the other for the stereo problem, we compared Graph Cuts and Belief Propagation on identical MRF's. The comparison was made

using the stereo framework created by Scharstein and Szeliski to compare a number of different stereo algorithms [10]. This framework can be found at `http://www.middlebury.edu/stereo`. In order to facilitate further experimentation, our implementation of the Belief Propagation algorithm and modifications to the stereo framework will be available at `http://www.ai.mit.edu/~mtappen`.

The MRF is defined in terms of energy functions, rather than compatibilities. The energy function $D(x_p, y_p)$, which corresponds to the matching cost computation and $\Phi(x_p, y_p)$, is computed using the Birchfield-Tomasi matching cost [2]. The cost function between nodes $V(x_i, x_j)$, which determines how support is aggregated and corresponds to $\Psi(x_i, x_j)$, is computed in the same fashion as [12]:

$$V(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ \rho_I(\Delta I) & \text{otherwise} \end{cases} \quad (4)$$

This type of energy function is known as a Potts model.

The function $\rho_I(\cdot)$ is defined in terms of the image gradient between the pixels i and j, which is denoted at ΔI:

$$\rho_I(\Delta I) = \begin{cases} P \times s & \text{if } \Delta I < T \\ s & \text{Otherwise} \end{cases} \quad (5)$$

where T is a threshold, s is a penalty term for violating the smoothness constraint and P is a penalty term that increases the penalty when the gradient has a small magnitude. Note that T, P, and s are constant over the whole image.

To use belief propagation, a cost C can be converted into compatibility by calculating e^{-C}. For numerical reasons, the cost is converted into a compatibility using $e^{-C/D}$, where D is a constant.

3.1. Choice of Belief Propagation Algorithm

To implement the Belief Propagation algorithm, two decisions must be made. First, either the sum-product algorithm or the max-product algorithm must be chosen. The sum-product algorithm computes the marginal distributions of each node, while the max-product algorithm computes the MAP estimate of the whole MRF. More information on these algorithms can be found in [6, 14, 16]. We use the max-product algorithm to find the MAP estimate for comparison with the Graph Cuts algorithm, which also computes the MAP estimate.

The second choice is the message update schedule. At each iteration, each node uses the messages it has received in the previous iteration from neighboring nodes to calculate messages to send to those neighbors. If node i is to the right of node j, node i sends a message to j at each iteration of the algorithm. This message contains node i's

belief about each possible state of node j. This message is computed from the messages that i has received from its neighbors. The message from i to j, denoted as $m_{right}(x_j)$ because it is the message that j is receiving from its right, is:

$$m_{right}(x_j) \leftarrow \max_{x_i} \Psi(x_i, x_j)\Phi(x_i, y_i) \times \\ m_{right}(x_i)m_{up}(x_i)m_{down}(x_i) \quad (6)$$

where $m_{right}(x_i)$, $m_{up}(x_i)$, and $m_{down}(x_i)$ are the messages received by i from the nodes above, below, and to its right.

The message update schedule determines when a message sent to a node will be used by that node to compute messages for the node's neighbors. In a synchronous update schedule, each node first computes the message for each neighbor. Once every node has computed the messages, the messages are delivered to each node and used to compute the next round of messages.

An alternative schedule is to propagate messages in one direction and update each node immediately. For instance the first node in a row, i would send a message to the node at its right, $i + 1$. Node $i + 1$ would then use this message immediately, along with the messages it had previously received from above and below, to compute a message to node $i+2$. Once this has been completed for every row, the same procedure occurs in the up, down, and left direction. We refer to this style of updating as "accelerated" updating.

The advantage of this method is that information is quickly propagated across the field. For a synchronous update schedule on an image with width W, it would take W iterations for information from one side of the image to reach the other. The alternative schedule would only require one iteration for this information to be propagated. This feature of the "up-down-left-right" message passing schedule causes the Belief Propagation algorithm to converge very quickly.

When the max-product algorithms converges on a graph with loops, it returns an approximate solution for the most likely labelling of the graph. The probability of this solution is guaranteed to be greater than all other solutions in a large neighborhood around that solution [15]. Upper bounds on the difference between the probability of the true MAP solution and the approximate solution returned by Belief Propagation are shown in [13].

3.2. Graph Cuts Algorithm

We used the Graph Cuts algorithm provided in Scharstein and Szeliski's package. In particular, the package implements the "swap" algorithm described in [5]. Like the Belief Propagation algorithm, the Graph Cuts algorithm

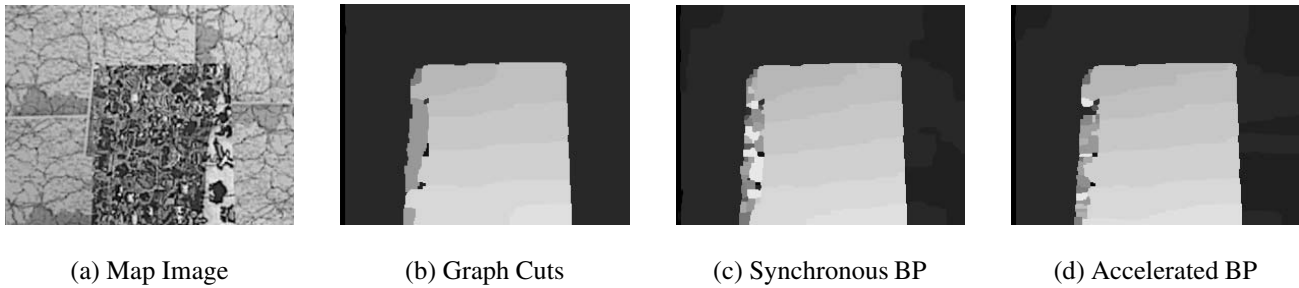(a) Map Image   (b) Graph Cuts   (c) Synchronous BP   (d) Accelerated BP

Figure 1. Results produced by the three algorithms on the map image. The parameters used to generate this field were $s = 50$, $T = 4$, $P = 2$. Graph Cuts returns the smoothest solution because it is able to find a lower-energy labelling than the two Belief Propagation algorithms.

	Energy of MRF Labelling Returned ($\times 10^3$)			% Energy from Occluded
Image	Ground-Truth	Graph Cuts	Synchronous Belief Prop	Matching Costs
Map	757	383	442	61%
Sawtooth	6591	1652	1713	79%
Tsukuba	1852	663	775	61%
Venus	5739	1442	1501	76%

Figure 2. Field Energies for the MRF labelled using ground-truth data compared to the energies for the fields labelled using Graph Cuts and Belief Propagation. Notice that the solutions returned by the algorithms consistently have a much lower energy than the labellings produced from the ground-truth, showing a mismatch between the MRF formulation and the ground-truth. The final column contains the percentage of each ground-truth solution's energy that comes from matching costs of occluded pixels.

finds a local minimum by making local improvements. The "swap" algorithm makes local improvements by choosing two of the possible states, α and β, then finding those nodes labelled α whose label should be change to β, or vice-versa, in order minimize the energy in the field as much as possible. Using the min-cut/max-flow formulation, the optimal swap for the entire graph can be computed.

4. Comparing Belief Propagation and Graph Cuts

We compared the Graph Cuts algorithm with the max-product Belief Propagation algorithm, using both synchronous updates and accelerated updates. For each of the four images used in [10], we generated 10 MRF fields by varying the T, s, and P parameters of Equation 5. We then used the Graph Cuts algorithm and the Belief Propagation Algorithms to estimate the MAP solution of the field. To compare the two algorithms, we collected the three statistics reported in [11] plus an additional statistic:

- $B_{\bar{\mathcal{O}}}$ – The percentage of pixels in non-occluded areas of the image with a disparity error greater than 1.

- $B_{\bar{\mathcal{T}}}$ – The percentage of pixels in textureless areas of the image with a disparity error greater than 1.

- $B_{\mathcal{D}}$ – The percentage of pixels near discontinuities in the image with a disparity error greater than 1.

- E – The energy of the solution.

4.1. Results for Map Image

The table in Figure 8 summarizes the results of the three algorithms on the map image. The performance in terms of $B_{\bar{\mathcal{O}}}$, $B_{\bar{\mathcal{T}}}$, and $B_{\mathcal{D}}$ is nearly identical; neither algorithm has a clear advantage.

However, it is useful to examine the energy of the solution returned by each algorithm. When the error penalty, s, is 20, the energy of the solutions returned by Belief Propagation and Graph Cuts nearly equal, although Graph Cuts consistently returns a smaller field energy. After s is raised to 50, the difference between the two solutions increases. The reason for this can be seen in Figure 1. The regions on the left side of the plane are smoother in the results returned by Graph Cuts than those returned by Belief Propagation.

However, this extra smoothness does not translate into better performance in terms of the ground-truth data. That
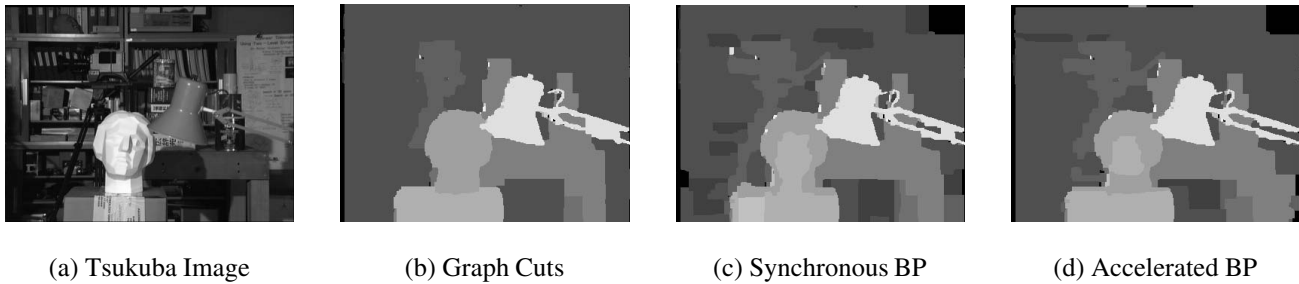
| (a) Tsukuba Image | (b) Graph Cuts | (c) Synchronous BP | (d) Accelerated BP |

Figure 3. Results produced by the three algorithms on the Tsukuba image. The parameters used to generate this field were $s = 50$, $T = 4$, $P = 2$. Again, Graph Cuts produces a much smoother solution. Belief Propagation does maintain some structures that are lost in the Graph Cuts solution, such as the camera and the face in the foreground.
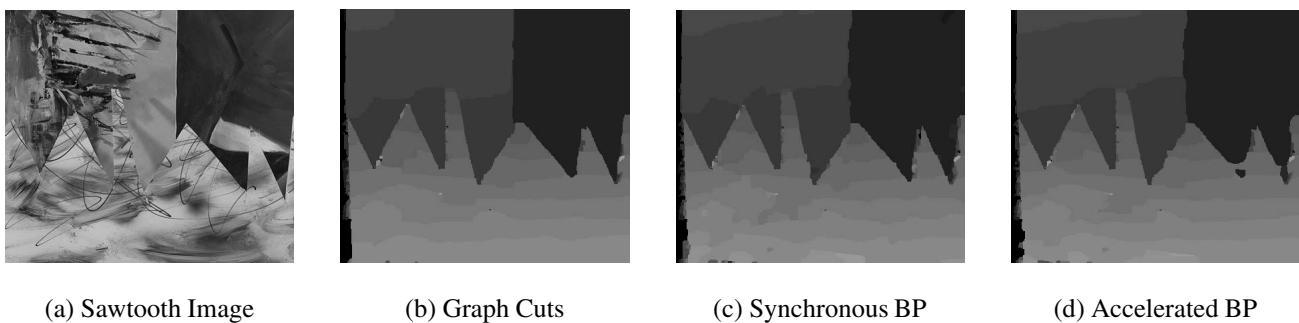

| (a) Sawtooth Image | (b) Graph Cuts | (c) Synchronous BP | (d) Accelerated BP |

Figure 4. Results produced by the three algorithms on the sawtooth image. The parameters used to generate this field were $s = 50$, $T = 4$, $P = 2$. For this image, the output of the three algorithms is comparable.

is because the ground-truth solution actually has a higher energy than either of the solutions returned by Belief Propagation or Graph Cuts. In Figure 2, the energy of the ground-truth solution for each image is shown for a specific setting of the parameters of $\rho_I(\cdot)$. The ground-truth labelling was produced by choosing the disparity level closest to the ground-truth disparity of each point. The energies of the labelling produced by Graph Cuts and Belief Propagation are significantly lower than the energy of the ground-truth labelling. The large energies for the ground-truth solution are caused by inaccurate matching costs in occluded areas. Since occluded pixels have no counterpart in the other image, the pixel at the correct disparity of an occluded pixel will likely have a different intensity, leading to a large matching cost. The significant effect of these matching costs can be observed in the last column of Figure 2. This column lists the percentage of the final energy for each of the solutions shown which can be attributed to matching costs for occluded pixels. These matching costs are a significant majority of the final costs.

4.2. Results for Tsukuba Image

The table in Figure 8 lists the results of the three algorithms on the Tsukuba image. For this image, Graph Cuts is superior. The primary reason for this superiority appears to be that the Belief Propagation algorithm assigns portions of the background to have very small disparity. An example of this can be seen in Figure 3. On the other hand, when the penalty, P, is higher, Belief Propagation does preserve some structures that Graph Cuts does not.

4.3. Results for Sawtooth Image

Figure 4 shows the output of the algorithm on the sawtooth image. In general, the results for the two algorithms on this image were comparable.

4.4. Results for Venus Image

Figure 5 shows a sample of the output of the algorithm on the venus image. Again, the Graph Cuts algorithm seemed to produce smoother results.

(a) Venus Image   (b) Graph Cuts   (c) Synchronous BP   (d) Accelerated BP
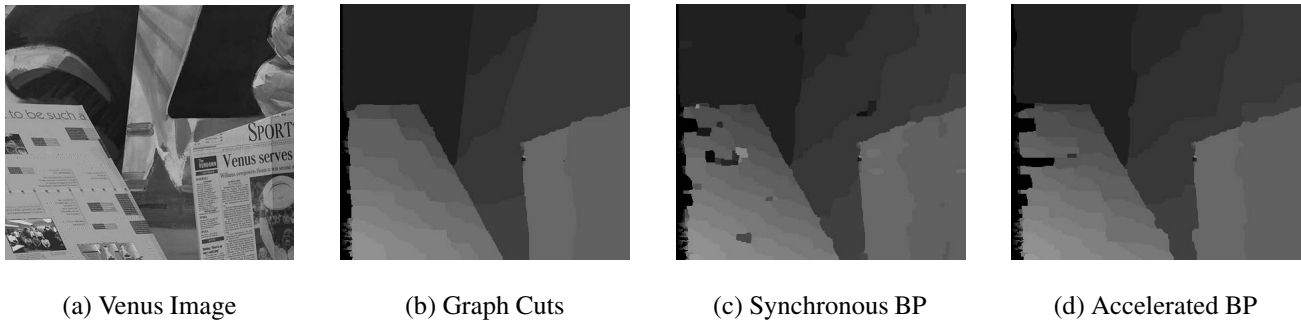
Figure 5. Results produced by the three algorithms on the venus image. The parameters used to generate this field were $s = 50$, $T = 4$, $P = 2$. Graph Cuts ability to find a lower energy solutions produces smoother, cleaner results.

Image	Time (sec.)		
	Graph Cuts	Synchronous Belief Prop.	Accelerated Belief Prop.
Map	268	1584	183
Tsukuba	268	1556	130
Sawtooth	272	3664	320
Venus	297	3644	262

Figure 6. The running times for the four examples shown. The accelerated Belief Propagation algorithm is generally the fastest. These results are from a 2.4 Ghz Pentium 4 processor.

4.5. Computational Time

In both belief propagation implementations, the algorithm ran for a fixed number of iterations. This number was chosen to be large enough that the algorithm converged for each of the images. We found that 50 iterations was enough for the accelerated message update scheme to converge. For synchronous updates, the number of iterations must be may need to be as large as the largest dimension of the image in order to pass information from one side of the image to the other. In practice, we found that setting the number of iterations to be equal to half of the largest dimension was suitable. Figure 6 shows the times for computing the four results examples shown. In general, the accelerated Belief Propagation algorithm takes less time to find a solution than Graph Cuts. It should be noted that there are faster versions of the Graph Cuts algorithm available [4]. The synchronous Belief Propagation algorithm takes significantly longer than the other two algorithms due to the large number of iterations required.

5. Discussion

5.1. Differences in Algorithm Performance

One of the interesting results is that the greatest difference between the performance of the Graph Cuts and Belief Propagation algorithms occurs when $s = 50$ and $P = 4$. In this case, the penalty incurred for neighboring derivatives with different disparities is either 50 or 200, depending on the strength of the image gradient at that point. The algorithms performed comparably when this penalty was smaller, such as when $P = 1$ or $P = 2$.

For Belief Propagation, the strong interaction penalties and tightness of the loops in the MRF likely hurt its performance. When applied to a network with loops, Belief Propagation returns an approximation to the optimal solution. This occurs because the loops allow the information propagated from the node to the rest of the MRF to eventually come back to the node. The strong interaction penalties in the MRF when $s = 50$ and $P = 4$ could exacerbate the effect of the loops.

The Graph Cuts algorithm's advantage in the case of the strong smoothness penalties may come from its construction. The max-flow/min-cut algorithm allows the Graph Cuts algorithm to compute the optimal swap for the whole graph. This global computation may help Graph Cuts cope with the strong interaction penalties better than Belief Propagation.

5.2. MAP vs MMSE

As described in Section 2.3, the MAP estimates were used for comparing Graph Cuts and Belief Propagation. A noticeable artifact of using the MAP estimate is stair-step effects, such as those seen in 7(a). This effect of having large flat regions with sudden jumps is caused because the MAP estimator must assign a single discrete disparity level to each point. Smoother depth maps can be obtained by instead using the MMSE estimator, which only Belief Propagation can compute. The MMSE estimate weights the discrete dispariy levels according to their marginal probability, resulting in sub-pixel disparities being assigned and a smooth depth-map, such as that shown in Figure 7(b). The marginal probabilities of each point can be found using the sum-product variant of the Belief Propagation algorithm, e.g. [6, 14].
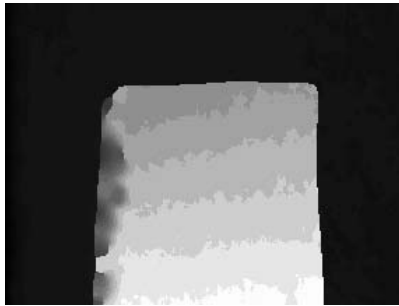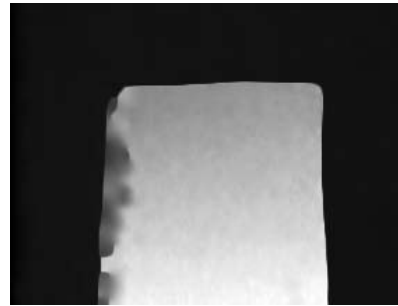
(a) MAP Estimate　　　　　　　　　　　　(b) MMSE Estimate

Figure 7. Comparison of MAP and MMSE estimates on a different MRF formulation. The MAP estimate chooses the most likely discrete disparity level for each point, resulting in a depth-map with stair-stepping effects. Using the MMSE estimate assigns sub-pixel disparities, resulting in a smooth depth map.

6. Conclusion

In this paper, we have examined how the Belief Propagation algorithm compares with the Graph Cuts algorithm when applied to the same Markov Random Field. We found that, in general, the results returned by the two algorithms were comparable, though the Graph Cuts algorithm was able to find lower energy solutions for the MRF's than Belief Propagation. Examining the output of the two algorithms, this was manifested in the fact that the solutions produced by Graph Cuts tended to be smoother. Neither the synchronous or accelerated Belief Propagation update schemes held a clear advantage over the other in terms of perfomance, while the acclerated Belief Propagation algorithm was significantly faster.

However, the smoother, lower energy solutions produced by the Graph Cuts algorithm did not necessarily translate into better performance compared to the ground-truth of the scene. This is because the solutions produced by both Graph Cuts and Belief Propagation always had a significantly lower energy than labelling the graph according to the ground truth. The best labelling of the pixels did not always correspond to the ground-truth. Given this situation, the greatest increase in performance will be found by improving the formulation of the MRF, rather than improving the solutions found for the MRF's currently being used.

For a stereo MRF using the Potts model, the Graph Cuts algorithm found lower energy solutions. However, we expect that the comparisons between these inference algorithms will vary as a function of the MRF formulation being solved. It would be useful to compare the expansion Graph Cuts algorithms with Belief Propagation using compatibility functions besides the Potts model, such as a truncated quadratic energy function.

Acknowledgments

References

[1] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Royal Statistical Society B*, 36:192–326, 1974.

[2] 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, April 1998.

[3] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 489–495, 1999.

[4] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. In *Third International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 2134 of *Lecture Notes in Computer Science*, pages 359–374. Springer, September 2001.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.

[6] W. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, 2000.

[7] B. J. Frey, R. Koetter, and N. Petrovic. Very loopy belief propagation for unwrapping phase images. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances*

IEEE
COMPUTER
SOCIETY

MRF Parameters			Algorithm											
			Graph Cuts				Belief Propagation (Accelerated)				Belief Propagation (Synchronous)			
T	s	P	$B_{\bar{\mathcal{O}}}$	$B_{\bar{\mathcal{T}}}$	$B_{\mathcal{D}}$	E	$B_{\bar{\mathcal{O}}}$	$B_{\bar{\mathcal{T}}}$	$B_{\mathcal{D}}$	E	$B_{\bar{\mathcal{O}}}$	$B_{\bar{\mathcal{T}}}$	$B_{\mathcal{D}}$	E
0	20	1	0.39	0	4.8	309574	0.37	0	4.9	314166	0.84	0	5.0	315455
4	20	2	0.35	1.4	3.9	313123	0.36	0	4.5	320731	0.41	0	4.9	322334
4	20	4	0.44	1.4	4.2	315770	0.46	0.95	4.83	323910	0.55	1.9	5.2	335110
8	20	2	0.34	0	4.5	320677	0.32	0	3.9	331511	0.36	0	4.7	336503
8	20	4	0.43	0	4.6	331302	0.39	0	3.82	347041	0.45	0	4.5	366189
0	50	1	0.20	0	3.8	366289	0.13	0	1.9	388394	0.14	0	1.9	385652
4	50	2	0.27	1.4	2.8	372693	0.15	0	2.0	400892	0.18	0.14	1.9	405933
4	50	4	0.29	0.14	2.6	375979	0.26	0.95	2.43	406647	0.71	0.95	2.2	427580
8	50	2	0.18	0	2.3	384342	0.14	0	2.0	417745	0.19	0	2.4	442665
8	50	4	0.39	0.4	2.9	399455	0.32	1.4	2.4	518615	0.38	1.9	2.9	518615

Figure 8. The results of applying the Graph Cuts and Belief propagation algorithms to the map image. Each algorithm returns the best results in terms of bad pixels for at least one setting of the parameters. However, graph cuts consistently returns the labelling with the lowest energy.

MRF Parameters			Algorithm											
			Graph Cuts				Belief Propagation (Accelerated)				Belief Propagation (Synchronous)			
T	s	P	$B_{\bar{\mathcal{O}}}$	$B_{\bar{\mathcal{T}}}$	$B_{\mathcal{D}}$	E	$B_{\bar{\mathcal{O}}}$	$B_{\bar{\mathcal{T}}}$	$B_{\mathcal{D}}$	E	$B_{\bar{\mathcal{O}}}$	$B_{\bar{\mathcal{T}}}$	$B_{\mathcal{D}}$	E
0	20	1	3.0	3.1	11	461041	3.4	3.1	11	477252	3.8	4.2	12	475829
4	20	2	2.1	1.3	10	489159	3.4	4.0	10	518809	3.1	2.9	11	526006
4	20	4	2.2	1.2	11	315770	3.9	4.8	11	576892	3.6	3.6	12	589790
8	20	2	2.0	1.1	9.8	503751	3.2	2.8	10	536803	3.0	2.9	10	538787
8	20	4	2.0	1.1	10	553215	2.98	3.01	9	604397	2.6	2.4	8	620444
0	50	1	2.9	3.0	13	602355	3.65	5.3	12	631680	3.3	4.4	12	633092
4	50	2	2.2	1.7	12	645865	3.4	4.8	10	711274	3.0	3.4	12	758117
4	50	4	2.7	1.7	13	696521	3.9	5.4	11	819633	2.5	2.7	9	899215
8	50	2	2.4	1.7	12	663845	3.5	4.6	11	726452	2.5	2.7	9	775085
8	50	4	2.8	1.7	14	739000	4.1	5.2	12	867552	2.8	3.2	9	941129

Figure 9. The results of applying the Graph Cuts and Belief propagation algorithms to the tsukuba image. Graph Cuts consistently performs better than Belief Propagation is all quality metrics, for this image.

in Neural Information Processing Systems 14, Cambridge, MA, 2002. MIT Press.

[8] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV (3)*, volume 2352 of *Lecture Notes in Computer Science*, pages 82–96, 2002.

[9] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *ECCV (3)*, volume 2352 of *Lecture Notes in Computer Science*, pages 65–81, 2002.

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

[11] J. Sun, H.-Y. Shum, and N.-N. Zheng. Stereo matching using belief propagation. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision - ECCV 2002*, number 2351 in Lecture Notes in Computer Science, pages 510–524, 2002.

[12] O. Veksler. *Efficient Graph-Based Energy Minimization Methods in Computer Vision*. PhD thesis, Cornell University, 1999.

[13] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. Tree consistency and bounds on the performance of the max-product algorithm and its generalizations. Technical report, Laboratory for Information and Decision Systems,MIT, 2002.

[14] Y. Weiss. Interpreting images by propagating bayesian beliefs. In *Advances in Neural Information Processing Systems 9*, pages 908–915, 1997.

[15] Y. Weiss and W. T. Freeman. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Trans. Information Theory, Special Issue on Codes on Graphs and Iterative Algorithms*, 47(2):723–735, 2001.

[16] J. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *International Joint Conference on Artificial Intelligence (IJCAI 2001)* Distinguished Papers Track, 2001.

IEEE
COMPUTER
SOCIETY