Efficient Stereo Algorithm using Multiscale Belief Propagation on Segmented Images

Hoang Trinh
Toyota Technological Institute at Chicago
1427 E. 60th Street, Chicago Illinois 60637, USA
ntrinh@tti-c.org
http://ttic.uchicago.edu/∼ntrinh/

Abstract

A variety of approaches using BP and image segmentation have been proposed for the stereo correspondence problem. In this paper, we introduce a novel approach, based on a combination of segmentation and BP. Our method inherits the idea of Multiscale BP, however at each level of the hierarchy, each graph node corresponds to an image segment, which we call superpixel, instead of a fixed rectangular block of pixels. The resulting depth map at the coarser level is used to initialize the depths at the finer level. At the lowest level, we perform loopy BP on the four-connected pixel subgrid within each superpixel. The proposed method is applied to stereo images in the standard Middlebury dataset, and to real outdoor stereo images and car sequences. Experimental results show quite acceptable accuracy of depth inference, with running time fast enough for practical use.

1 Introduction

Stereo vision has for many years been considered a fundamental problem in computer vision, and still continues being an active research topic now. A substantial amount of work has been developed to solve the stereo matching problem [10]. Among these proposed methods, Belief Propagation (BP) and segmentation-based methods have been widely investigated. BP for stereo matching has been described in [7, 5, 8, 9] and some other papers. Some recently introduced segmentation-based methods [3, 2, 1, 11] have obtained very good performance on the Middlebury dataset.

The approach we introduce in this paper combines segmentation and BP. Our method is inspired by the idea of Multiscale BP [5]. The general loopy BP algorithm is too slow for practical use as a large number of iterations is usually a necessary condition for convergence, and each BP iteration has to run across the whole image grid. The Multiscale BP algorithm described in [5] is an algorithmic technique that improves the efficiency of BP, where BP is performed in a coarse-to-fine manner, so that long range interactions between pixels can be captured by short paths in coarse graphs. In Multiscale BP, the graph structure is constructed as follows. The graph at the i-th level consists of blocks of $2^i \times 2^i$ pixels, connected in a grid structure. Therefore the graph at the next level is a finer grid of the graph at the previous level. Our MRFs are built in a totally different way. At the top level, each graph node corresponds to an image segment, which

we call a superpixel, instead of fixed rectangular blocks of pixels. Adjacent superpixels are connected by edges in the graph. Loopy BP is then performed on this graph to assign a depth to each superpixel. At the next level, each superpixel is further segmented into smaller superpixels. Again, we build a graph on top of these superpixels and perform BP for depth inference. The resulting depth of the parent superpixel at the coarser level is used to initialize the depths of children superpixels at the finer level. At the lowest level, we perform BP on the four-connected pixel grid within each superpixel.

This significant difference in implementation brought to our approach some distinguishing properties. First, it is a natural way to use the superpixels acquired from the segmentation as pixel blocks for the multiscale approach. Our pixel blocks therefore actually correspond to real objects or object parts in the images, while the square blocks in Multiscale BP are simply created for the sake of computational convenience, and do not capture any perceptual meaning of the image data. As a consequence, we can allow for a sharp discontinuity of depth between superpixels, while sill maintain the smoothness assumption within each superpixel. The effect this has is that our approach is less likely to suffer from the over-smoothing problem, which causes the blurriness at thin objects, and at depth discontinuities. The general loopy BP algorithm as in [7, 5, 8, 9] and others, is a global stereo method, which incorporates explicit smoothness assumptions throughout the image and determines the disparity map by minimizing a global energy function. This behavior of propagating the smoothness cost information across the whole image may lead to poor performance at object boundaries.

Our approach shares the same assumption with other segmentation-based methods [3, 2, 1, 11], i.e. the scene structure can be approximated by a set of non-overlapping planes in the disparity space, and each plane corresponds to one segment in the image. However most of those methods are too complicated and therefore become too slow for practical applications. Another main distinction of our approach is that segmentation-based methods generally perform a local-based matching step to detect a set of reliable point correspondences. This step depends heavily on the image data, especially the color information, and is likely to perform poorly on less idealized real-world data (including, for example, gray-scale images or night-vision images, which are often noisy, with less contrast and of lower resolution). Our approach uses loopy BP to minimize a global cost function, which takes into account both the data and the spatial coherence, therefore is more robust to noisy data.

For the purpose of performance evaluation, we tested our algorithm with the standard Middlebury dataset. The testing results in comparison with the Multiscale BP algorithm are demonstrated. Since we are more interested in the performance of our method in practical applications, such as stereo vision for autonomous vehicle systems, we also apply our method to the real outdoor stereo car sequence provided by Daimler AG and Toyota. Experimental results show quite acceptable accuracy, with encouraging running time.

In the next section of the paper, we describe our proposed algorithm in details. Section 4 demonstrates experimental results obtained from the standard stereo image dataset with associated ground truth, as well as from real stereo car sequences. The last section includes some discussions and conclusions.

2 Proposed approach

In this section, we present a detailed description of our approach for the stereo matching problem.

2.1 Segmentation and Graph construction

To segment the image, we use an efficient algorithm described in [4]. This algorithm has advantage over the color-based segmentation commonly used by other segmentation-based stereo methods. It captures perceptually important non-local image regions, which often reflect global aspects of the image. This property strongly supports the assumption that the scene structure can be approximated by a set of non-overlapping smooth surfaces in 3D space, and each surface corresponds to an image segment/superpixel. In addition, this segmentation algorithm runs in time nearly linear in the size of the image and is fast in practice.

Our algorithm also requires a construction of a graphical models between superpixels at each segmentation scale. This graph is built by assigning each node to a superpixel and adding an edge between each pair of adjacent superpixels. In order to do this, we need to capture the adjacency relations between superpixels. As the algorithm in [4] is based on an iterative region merging technique, in which neighboring image regions with similar characteristics are merged to form a new region, it turns out that we can follow each iteration of segmentation and keep track of the adjacency information between superpixels. Therefore, segmentation and graph construction can be done simultaneously, and run in time linear to the size of the image.

It is also important to note that except for the first scale, there are more than one graph at all other scales. In fact, the number of MRFs at the next scale is exactly the number of superpixels at the previous scale. At the next scale, each superpixel of the previous scale is further segmented into a set of smaller superpixels. An MRF is then formed for each set and loopy BP is performed on this MRF for depth inference. This procedure is illustrated in Figure 1. The smoothness constraint across superpixels at a specific scale is only enforced at that scale and ignored at subsequent scales. The reason to do this is based on the following argument. The property of the segmentation algorithm that we use almost guarantees that boundaries between superpixels are aligned with real boundaries of different surfaces in the real world. This allows for a truncation of depth smoothness at boundaries of superpixels. This truncation also helps reduce a large number of passing messages. The number of messages being truncated at each scale is exactly the total length of boundaries between superpixels at the previous scale. Although in our implementation, depth inferences in different MRFs at one scale are performed sequentially, we can also run them in parallel to make the algorithm much more efficient.

2.2 Loopy BP for depth computation

First we introduce several definitions and notations that will be used later in this section. The cost of assigning a depth d to a pixel (x,y) is defined as follows:

$$D(x,y,d) = D_I(x,y,d) + \omega * D_{GRA}(x,y,d)$$

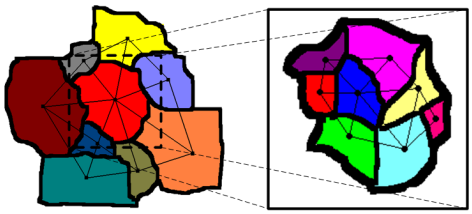


Figure 1: A superpixel at the previous scale and its corresponding MRF at the next scale.

where ω controls the relative weight between the 2 terms. $D_I(x,y,d)$ measures the squared intensity differences, and is given by

$$D_I(x,y,d) = (I_{left}(x,y) - I_{right}(x-d,y))^2$$

$D_{GRA}(x,y,d)$ incorporates the squared gradient differences in both directions x and y, which makes $D(x,y,d)$ more robust to change of brightness and angle of views:

$$D_{GRA}(x,y,d) = (\nabla_x I_{left}(x,y) - \nabla_x I_{right}(x-d,y))^2 + (\nabla_y I_{left}(x,y) - \nabla_y I_{right}(x-d,y))^2$$

In case of color images, these dissimilarity measurements can easily be extended to all channels. The cost of assigning depth d to a superpixel s is defined as follows:

$$D_{sp}(s,d) = \sum_{(x,y)\in s} D(x,y,d)$$

Next, the cost of assigning depth d_i and d_j to two neighboring pixels i and j is defined to be $V(d_i,d_j) = |d_i - d_j|$. This term formulates the spatial coherence constraint at the pixel level. For the superpixel level, the term becomes $V_{sp}(s_i,s_j,d_i,d_j) = \sigma |d_i - d_j|$, where σ is a weighting function that involves the squared difference in mean intensity of the two superpixels s_i and s_j. The idea is that the smoothness violation between adjacent superpixels should be penalized more for superpixels that look similar, and less for superpixels that look different.

Now we describe our segment-based multiscale BP approach. For each MRF G, we search for an optimal assignment of depth $d_s \in D_G$ to each superpixel $s \in G$. (in the first segmentation scale, G covers the whole image). D_G is the total number of quantized depth values that can be assigned to nodes in G. This amounts to optimizing an energy function for the labeling f that labels each superpixel s with a corresponding depth $d_s = f(s)$. The energy function has the following form:

$$E^G(f) = E^G_{data}(f) + \lambda * E^G_{smooth}(f)$$

where λ controls the relative importance of the data cost and the smoothness cost,

$$E^G_{data}(f) = \sum_{s\in G} D_{sp}(s, f(s))$$

and

$$E^G_{smooth}(f) = \sum_{\forall (s_i,s_j)\in N_G} V_{sp}(s_i,s_j,f(s_i),f(s_j))$$

where N_G is the set of all pairs of neighboring superpixels in G. The optimal depth assignment for all nodes in G is approximated by performing loopy BP on G. We used the max-product BP algorithm with conceptually parallel updates. In our actual implementation however, they were performed sequentially.

Due to the difference in the graph structure, the messages cannot be passed explicitly from one scale to the next, as in Multiscale BP. Here the information from coarse to fine is softly transfered downwards as follows. First, each message coming to a child superpixel will be initialized to be biased toward the depth value assigned to its parent superpixel. Second, as we assume that each superpixel is a region with small depth variability, the depth values of the children superpixels can be bounded within a small range around the depth value that was assigned to their parent superpixel. As a result, the depth range used for children superpixels can be much smaller than the one used for the parent superpixels at the previous scale. Since the messages at the fine level here are less dependent on the messages at the coarse level, our method tends to be less sensitive to errors made at the coarse level.

At the final scale, loopy BP is performed on the pixel grid of each superpixel at the previous scale and a depth value is assigned to each pixel in the image. In this case G is a subgraph of the four-connected pixel grid, and each term in the energy function now becomes:

$$E^G_{data}(f) = \sum_{p \in G} D(p, f(p))$$

where p is a pixel in G and

$$E^G_{smooth}(f) = \sum_{\forall (p,q) \in N_G} V(f(p), f(q))$$

where (p,q) are pairs of neighboring pixels in G.

Here we performed a significant pruning of smoothness terms from one level to the next, especially at the lowest level, where the smoothness constraint is only enforced between the pixels inside a superpixel. As discussed in section 1, boundaries between superpixels are very likely to be aligned with real boundaries of objects in the real world. This means that the set of region boundaries will cover most occlusion boundaries. It is quite valid to truncate the smoothness across these boundaries. Conversely, a lot of region boundaries are not occlusion boundaries. The concern is that the pruning of valid smoothness terms at these boundaries may result in noisy stereo estimates. Specifically, depth discontinuities may incorrectly occur at region boundaries which are not occlusion boundaries. The problem may get more severe in images containing large object surfaces. For example in Figure 4 and Figure 5, there exist visible depth discontinuities inside the region of the ground plane. In this case, the smoothness truncation can be considered a trade-off between quality and speed. Here we argue that for many cases this smoothness truncation is completely acceptable. First, note that we still enforce the smoothness constraint between superpixels at the previous level. Second, the depth information is transfered from coarse to fine as previously described. The impact is that nearby pixels at the lowest level are still driven to have similar depths, i.e the smoothness constraint between pixels are still implicitly enforced. Consider the images in Figure 2: Although most region boundaries created by the segmentation are not occlusion boundaries, these boundaries are not visible in the resulting depthmaps, while the true occlusion boundaries were well maintained.

3 Experimental Results

In this section, in order to evaluate the performance of our algorithm, the following experiments were carried out on stereo dataset with ground truth, and on real-world images and car sequences. For all datasets we used a fixed set of parameters ω and θ.

3.1 Qualitative Analysis with the Middlebury dataset

We evaluate our algorithm on the Middlebury dataset[1], where 4 stereo pairs and their ground truth are provided. Figure 2 shows the obtained results, compared to the ground truth. We also compare our method with the Multiscale BP algorithm in [5]. Since our method is based on this method, it is meaningful to show that our method can improve its performance. Here we used the publicized source code from Pedro Felzenszwalb[2]. We kept the default values for all parameters used in their implementation. They had better results on the first 2 categories: *tsukuba* and *venus*(with depth ranges of 16 and 20, respectively), while we outperformed on the last 2 categories: *teddy* and *cones* (with depth range of 60). Note that the scenes in the last 2 categories are more complicated, and have broader depth ranges. The comparison is illustrated in table 1.

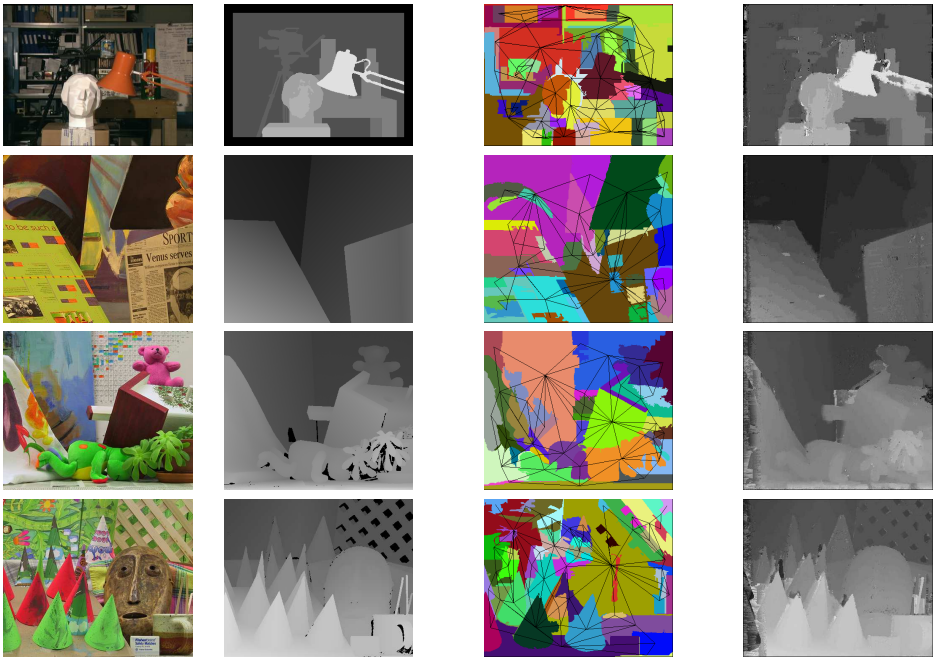


Figure 2: Results using the Middlebury dataset with available ground truth.

[1]The dataset can be downloaded from http://cat.middlebury.edu/stereo/data.html
[2]The source code is available at http://people.cs.uchicago.edu/ pff/bp/.

	Tsukuba			*Venus*			*Teddy*			*Cones*		
	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
MBP[5]	2.13	4.29	11.4	1.40	2.38	16.5	17.3	25.2	31.0	12.5	20.6	22.0
Our method	4.85	7.03	19.0	7.40	8.94	28.1	14.2	22.8	30.9	10.6	20.5	22.9

Table 1: Comparison between the proposed method and the Multiscale BP algorithm on the *Middlebury* dataset (error threshold = 1).

3.2 Results on realistic stereo images and sequences

Real-world data are different from standard database in that they usually have lower quality (for example, outdoor gray-scale images or night-vision images, which are often noisy, with less contrast and of lower resolution), and have a much larger depth range, as well as more complicated scene. The objective of this experiment is to evaluate the performance of our method on such data. Unfortunately ground truth depth is not available for these datasets.

Figure 3: Results using some realistic images.

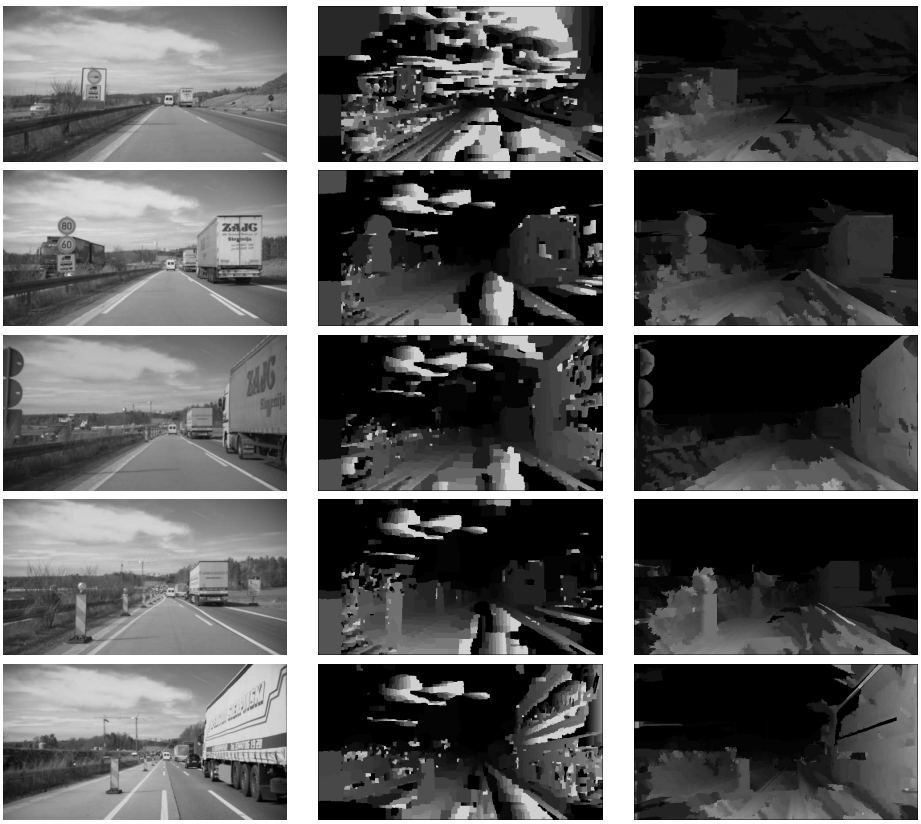
Figure 3 demonstrates our results on several real-world grayscale images. We also tested our method on the real-world road driving rectified stereo sequences (acquired and provided by Daimler AG[3] and Toyota). The resulting depthmaps of a few frames in the sequences are demonstrated in figure 4 and 5. By means of comparison, results from the Multiscale BP algorithm were also shown.

For running time analysis, we run our algorithm for the 2 car sequences and compute the average running time on each frame. All the frames in both stereo sequences are grayscale and have been geometrically rectified. We then compare our running time with the Multiscale BP algorithm (Table 2). Both algorithms use the same testing environment as follows: laptop PC with Intel Core Duo 2.0Ghz, 2 Gigabyte memory, on-board Mobile Intel(R) graphic adapter, WinXP operation system. Both uses the depth range of 53 pixels. Multiscale BP runs with 10 iterations, while our algorithm runs BP on 3 scales, each with 10 iterations.

4 Discussions and Conclusions

In this paper we introduced a new algorithm for stereo vision based on the conjunction of loopy BP and image segmentation. Similar to other segmentation-based methods, our ap-

[3]The sequence is available at http://www.citr.auckland.ac.nz/6D/datasets.htm



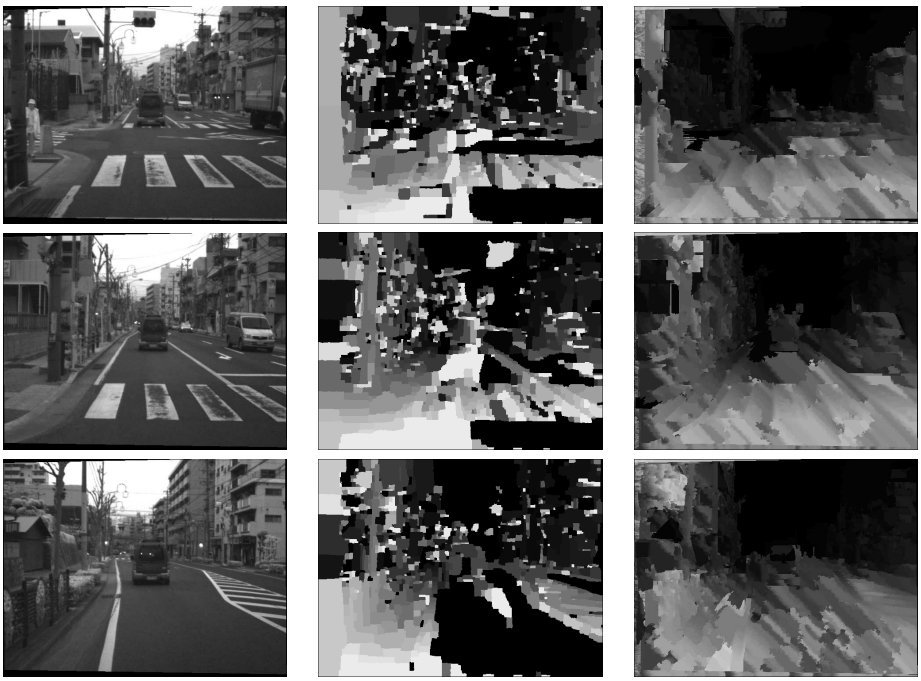
Left image Multiscale BP Our results

Figure 4: Results using the Daimler road driving sequence.

proach uses the assumption that each superpixel in the image corresponds to a smooth surface of the scene structure, and these surfaces do not overlap. Due to this assumption, the piecewise smoothness prior is automatically incorporated within each superpixel, while the smoothness constraint across superpixels is enforced by running loopy BP at each level of multiscale segmentation. Depth assignment by Loopy BP is performed downwards at each scale of segmentation, starting from a coarse segmentation of the image, and the lowest level being the pixel grid within each superpixel of the most recent segmentation.

By using the superpixels acquired from the segmentation as pixel blocks for the multiscale approach, each pixel block at each scale actually represents a smooth surface of the scene structure. Consequently, depth discontinuity is allowed at boundaries of superpixels, while we still assume smoothness within each superpixel. This property not only reduces significantly the number of passing messages, but also helps maintain sharp depth discontinuities at object boundaries, and maintain thin objects.

The proposed method is then applied to the standard stereo dataset as well as to the outdoor stereo car sequences provided by Toyota and Daimler AG. With the Middlebury dataset, we obtain comparable performance with the standard loopy BP algorithm. As we



Left image Multiscale BP Our results

Figure 5: Results using the Toyota night-vision road driving sequence.

want to give emphasis to practical applications of stereo vision, we are more interested in evaluating our methods on real-world data. Experimental results, as compared with the Multiscale BP algorithm, show that we have more encouraging running time, with better performance on the car sequences. In this data, we especially do better on low-textured regions (sky, ground planes, ...), and we also obtain sharper boundaries in the resulting depth map.

One problem with this method is that it is sensitive to error made at the coarse level, since it is strictly coarse to fine. We can address this problem by constructing a graphical model that models mutual relations between superpixels at different levels. One possible choice would be to connect the MRFs at all levels to form a unified MRF, by adding an edge between the parent superpixel at one level and each of its children superpixels at the next level. However, this would add a lot more complexity to the system, and would

	Avg running time	
	Daimler (300 frames) 640×350 *pixel*	Toyota (400 frames) 644×493 *pixel*
Multiscale BP	19.5s/fr	25.5s/fr
Our method	16.3s/fr	18.4s/fr

Table 2: Average running time of our method and the Multiscale BP on the car sequences.

reduce the significantly the efficiency of the algorithm.

In the near future, we plan to work on speeding up the algorithm. The most obvious approach is to run the loopy BP algorithm at each MRF in a parallel fashion. Also, the same algorithm can be implemented incorporating a technique to handle occlusions, and a more sophisticated 3D plane representation for each superpixel, in order to improve accuracy.

References

[1] M. Sormann A. Klaus and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR*, 2006.

[2] M. Bleyer and M. Gelautz. A layered stereo algorithm using image segmentation and global visibility constraints. In *ICIP*, 2004.

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

[4] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2), September 2004.

[5] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1), October 2006.

[6] M. Gong and Y.-H. Yang. Near real-time reliable stereo matching using programmable graphics hardware. In *CVPR*, 2005.

[7] N.N. Zheng J. Sun and H.Y. Shum. Stereo matching using belief propagation. *PAMI*, 25(7):787–800, 2003.

[8] R. Yang H. Stewénius Q. Yang, L. Wang and D. Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. In *CVPR*, 2006.

[9] R. Yang S. Wang M. Liao Q. Yang, L. Wang and D. Nistér. Real-time global stereo matching using hierarchical belief propagation. In *BMVC*, 2006.

[10] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2002.

[11] L. Zitnick and S.B. Kang. Stereo for image-based rendering using image over-segmentation. *IJCV*, 2007.