Near Real-time Stereo for Weakly-Textured Scenes

Qingxiong Yang[1] Chris Engels[2] Amir Akbarzadeh[3]

[1]Department of Electrical and Computer Engineering
University of Illinois at Urbana Champaign
qyang6@uiuc.edu

[2]ESAT-PSI/VISICS
K.U. Leuven
cengels@esat.kuleuven.be

[3]Microsoft Live Labs
amir@microsoft.com

Abstract

Several real-time/near real-time stereo algorithms can currently provide accurate 3D reconstructions for well-textured scenes. However, most of these fail in sufficiently large regions that are weakly textured. Conversely, other scene reconstruction algorithms assume strong planarity in the environment. Such approaches can handle lack of texture, but tend to force nonplanar objects onto planes. We propose a compromise approach that prefers stereo depth estimates but can replace estimates in textureless regions with planes in a principled manner at near real-time rates. Our approach segments the image via a novel real-time color segmentation algorithm; we subsequently fit planes to textureless segments and refine them using consistency constraints. To further improve the quality of our stereo algorithm, we optionally employ loopy belief propagation to correct local errors.

1 Introduction

Our paper presents a robust method for correcting textureless areas in stereo depth maps using locally estimated planes. The approach is especially relevant to 3D reconstruction of urban and other man-made scenes, for which many areas in an image may contain planar objects. We designed our system with an emphasis on performance in order to facilitate the computation of large reconstructions gathered from video sequences.

When creating reconstructions from video, small subsequences can be reconstructed via stereo depth estimates, which are later combined into a larger reconstruction. While accurate depth estimates are important for generating usable reconstructions, the best current algorithms run offline. Since the number of depth maps necessary for reconstructing an urban area is typically high, allowing individual estimates to run until some optimal convergence or otherwise execute for an arbitrarily long time may not be feasible, so offline algorithms are not typically appropriate here. Several online approaches also exist, often executing on graphics hardware or on low resolution images. However, such approaches often fail in large textureless or weakly-textured regions, since stereo correspondence is uninformative. Unfortunately, urban areas tend to have many such regions.

BMVC 2008 doi:10.5244/C.22.72

Conversely, other reconstruction approaches forgo dense stereo estimation by assuming image regions correspond to facades, which can be approximated as planes. Such approaches can handle textureless areas well, since they can simply be assigned to nearby planes. The resulting models are clean, possibly to the point of oversimplification, since they may miss finer depth details that stereo estimation can reconstruct.

We employ a philosophically different approach to plane fitting from those discussed at the end of Section 2, which typically assume any observed region in an image lies on or close to a plane. In well-textured areas, stereo depth estimates are trustworthy and reflect observed data more precisely (*i.e.* at pixel and even subpixel scales) than large scale plane fitting. We choose to trust the observed data as much as possible and only attempt to fit planes to regions that are locally uninformative and therefore can not provide useful depth cues independently.

In this paper, we propose a near real-time plane-fitting stereo pipeline to deal with this problem. Our pipeline contains three modules: window-based multi-view stereo matching, stereo fusion and plane-fitting refinement. To achieve high performance, we propose a novel real-time color segmentation approach in the last module. We also propose an optional modification based on belief propagation (BP), where after stereo matching, an initial plane-fitted depth map is created, followed by a loopy belief propagation refinement. This addition helps to correct potential errors caused by plane-fitting due to non-robustness of color segmentation.

We discuss previous work in Section 2. Section 3 provides an overview of the algorithm that provides a basis for our approach. Sections 4 and 5 detail our improvements via plane fitting and belief propagation, respectively. Results are shown in Section 6, and Section 7 concludes.

2 Related Work

Several surveys of a number of classes of stereo algorithms are given in [4, 14, 15]. As stated in [14], local stereo algorithms are dependent on their aggregation windows. If a local algorithm encounters a textureless area larger than the aggregation window, i.e. the depth estimate for a given pixel has no unique support within a local region, the algorithm is guaranteed to fail. Moreover, most real time algorithms are local, meaning that in weakly-textured environments, such algorithms will produce large amounts of error.

Global algorithms, such as graph cuts [3] and belief propagation [16], have properties that can improve depth estimates in difficult environments. These algorithms rely on minimization of some global cost function. In textureless areas, the minimization tends to have a smoothing or blurring effect. Because of their iterative nature, they are typically too slow for limited time frame applications.

Instead of attempting to infer depth purely from stereo matching, several methods exploit the planarity implicit to urban environments. Baillard and Zisserman [2] use a 3D line and surrounding texture to hypothesize planes in an image. Similarly, Werner and Zisserman [17] automatically search for scene planes in a set of images using point and line correspondences. Since the authors focus on architectural scenes, they assume most of the reconstruction will be limited to a few dominant planes and compensate for deviations from this assumption as a secondary step. Cornelis *et al.* [6] describe a real-time method for creating simplified urban models that assumes all surfaces are either on

a ground plane or a plane orthogonal to it. The planar prior modifies the cost function, so instead of choosing the depth with the lowest cost over an aggregation window, costs for depths close to the prior become slightly lower. Their approach relies on the ability to robustly calculate the correct plane priors from the sparse structure, which can be difficult in many scenes.

In contrast to these methods, we place no initial assumption of planarity on the scene and use plane estimation only as an error compensation method for depths we can not otherwise determine.

3 Window-based Stereo

A real-time local window-based stereo pipeline is described in [1], which we briefly review here.

The primary step of a multi-view stereo matching module is the plane-sweeping algorithm of [5]. Given a sequence of consecutive images, the depth map is computed for the central image, denoted the reference image. A set of planes is swept through space at a number of hypothesized depths. Each plane defines a set of homographies with which all the non-reference images are warped onto the reference image. The absolute intensity difference defines a cost for each hypothesized depth. The set of images is divided in two halves, one preceding and one following the reference image. The costs are aggregated by a boxcar filter and the minimum of the two sums defines the cost of the depth hypothesis [11] (this is an effective way of handling occlusions). Unfortunately, the hypothesis having the lowest cost may not always be the true depth. This is in most cases due to lack of texture. Therefore a confidence map is needed to denote how certain we are about each chosen depth hypothesis. We follow the suggestion of Merrell *et al.* [13] and define our stereo confidence function C_s as

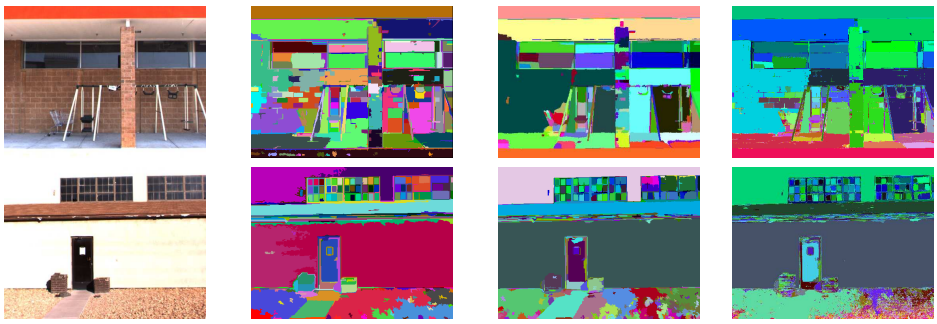
$$C_s(p) = \left(\sum_{d \neq d_{est}} exp(-(c(p,d) - c(p,d_{est}))^2 / \sigma^2) \right)^{-1}, \quad (1)$$

where p is a pixel, $c(p,d)$ denotes the matching cost at a depth d, d_{est} is the estimated depth, and σ is a constant dependent on noise.

Given a sequence of consecutive depth maps, we next enforce consistency among these maps and output an improved set of depth maps as in [13]. However, even after this step there may still be pixels for which the depth estimate is unlikely or wrong, so each new depth map is again associated with a confidence map. We compute this new fused confidence map, denoted C_f, by adding the confidences corresponding to depth estimates that were consistent within some interval to the fused depth estimate for each pixel.

4 Plane-fitting Stereo

The Plane-fitting stereo pipeline begins with window-based stereo matching and consistency fusion as described in Section 3. We then apply a novel real-time color segmentation approach, where a plane is fit for each output segment in order to obtain correct depth values for the weakly-textured regions.



(a) Color image. (b) Graph-Based [8]. (c) EDISON [7]. (d) Our approach.

Figure 1: Comparison of different color segmentation approaches. For one 512×384 image, (b) requires 0.5 seconds, (c) 4 seconds, and (d) 0.067 seconds.

4.1 Real-time Color-weighted Color Segmentation

We separate our segmentation approach into two steps: image smoothing and region linking. To preserve the edges, we use a color-weighted filter [18, 21] to smooth the image. The support from a neighboring pixel q to a pixel under consideration p is weighted as

$$w(p,q) = exp(-(\frac{\triangle c_{pq}}{\gamma_c} + \frac{\triangle s_{pq}}{\gamma_s})), \tag{2}$$

where $\triangle c_{pq}$ is the maximal color difference between p and q measured in each channel of the CIELUV color space and $\triangle s_{pq}$ is the distance between p and q in the image domain, and γ_c and γ_s are weighting parameters. To achieve real-time performance, the color space transformation and the smoothing steps are done in Graphics Processing Unit (GPU), and the range of the three channels of CIELUV color space is confined in $[0,255]$. We apply the filter five times, with $\gamma_c = 2.0$ and $\gamma_s = 10$ determined experimentally. This smoothing step processes about 15 frames per second (fps) on 512×384 resolution images using our GPU implementation.

We next link all 8-connected pixel neighbors p and q if $\triangle c_{pq} < \gamma_c$. The region linking processes at 33 fps on CPU. The smoothing step remains the bottleneck, and the overall segmentation performance is 15 fps.

Figure 1 provides a visual comparison of our segmentation approach with two other approaches. Our real-time segmentation approach is comparable to the other two algorithms, although some over-segmentation occurs in heavily-textured areas. However, we would expect that stereo matching will perform well in these areas, so over-segmentation will not adversely affect our approach.

4.2 Plane-fitting

The goal of plane-fitting is to correct depth values that we believe to be incorrect, for example depth estimates computed in weakly textured image regions. We classify all the pixels in the reference depth map into stable and unstable pixels by setting a threshold for the confidence map C_f. For each selected segment S_I^j in the reference image, we robustly

fit a 3D plane using a RANSAC approach [10] on the depth values of the stable pixels only. We back project all stable pixels $p_k \in S_I^j$ to 3D world points $P_k \in S_W^j$. A set of hypothesis planes are generated by randomly selecting three 3D points and computing the plane that intersects these. The vector defining the plane is then normalized and each plane is associated with the following error cost:

$$E_\pi(S_I^j, \pi_j) = \sum_{P_k \in S_W^j} \min(P_k^T \pi_j, \eta_d), \tag{3}$$

where η_d is a constant to increase robustness by bounding the penalty of potential outliers. Finally, the plane hypothesis with the minimum cost is selected and the depth values of *only the unstable pixels* are replaced with the plane-fitted depth values. Only sufficiently large segments will be fit with a plane, since a small segment suggests variations in the segments neighborhood, i.e high texture.

The plane-fitting approach may fail if the number of stable pixels in the segment is too small. In this case, we compute a bounding box containing the segment and instead use all the stable pixels within the bounding box for the RANSAC plane-fitting.

In order to remove small differences between plane fitted unstable pixels and the original stable pixels, we add an adaptive smoothing step. We replace each depth value with the average of those values that are within a threshold $\sigma_p < 0.5$ over a 9×9 window.

Finally, we apply a consistency check to the plane-fitted depth maps in order to reject outliers using a new confidence map defined as follows:

$$C_c(p) = h(\sum_{i=1}^{N} h(|D_i'(p) - D_{ref}(p)|, \sigma_c), \eta_c), \tag{4}$$

where

$$h(a,b) = \begin{cases} 1 & \text{if } a >= b \\ 0 & \text{else} \end{cases}.$$

D_i' is one of the N neighbouring depth maps projected onto the reference depth map, D_{ref}. $\sigma_c = 0.2$ and $\eta_c = N - 1$ are thresholds for the consistency check. This confidence map is passed on to the mesh generation module where triangles are only created for depth values that have a high confidence.

5 BP-based Plane-fitting Stereo

We now describe a modification to the first module of our pipeline incorporating BP-based stereo matching. While not sufficient for correcting large areas, a few iterations of belief propagation can help to correct potential fattening and robustness errors caused by the stereo matching, color segmentation, and plane fitting steps. The last two modules are the same as the plane-fitting stereo pipeline described in Section 4.

After window-based stereo matching, the pixels are classified into stable and unstable pixels based on the confidence map as described in Section 3, after which plane-fitting is computed for large segments as described in Section 4.2. To correct the potential errors, a GPU hierarchical loopy belief propagation approach is implemented according to [19].

The loopy belief propagation minimizes the following energy function:

$$E(p,d) = E_D(p,d) + E_S(p,d), \tag{5}$$

where p is a pixel and d is the depth hypothesis.

The data term $E_D(p,d)$ is constant and is defined as:

$$E_D(p,d) = C_s(p)\min(E_m(p,d),\eta_m) + (1-C_s(p))\min(\beta(d-D_\pi(p))^2,\eta_\pi), \quad (6)$$

where $C_s(p) \in [0,1]$ is the confidence map calculated from the correlation volume, E_m is the correlation volume without boxcar aggregation, and D_π is the plane-fitted depth map. By integrating C_s, E_m and D_π, the data term E_D depends mostly on the plane-fitted depth map in the low confidence areas and on the correlation volume in the high confidence areas. The constant $\eta_m = 50.0$ is used to reject outliers in the correlation volume. $\beta = 2.0$ is the rate of increase in the cost caused by the plane-fitted depth map D_π and $\eta_\pi = 50.0$ controls when the cost stops increasing.

The smoothness term $E_S(p,d)$, which is based on the assumption that the world surfaces are piecewise smooth, is iteratively minimized by passing messages to p from its neighbors, which we form similarly to [9]. The message passed from q to p at iteration i is defined as

$$M^i_{q\to p}(d) = \underset{d_q}{\operatorname{argmin}}(E_D(q,d_q) + \sum_{s\in N(q), s\neq p} M^{i-1}_{s\to q}(d_q) + E_j(d_q,d)), \quad (7)$$

where $N(q)$ is the four-connectivity neighborhood of q, $E_j(d_q,d)$ is the jump cost, and d is the label that minimizes the total energy for pixel q, which contains the data term and the smoothness term. The jump cost $E_j(d_q,d)$ is based on the degree of difference between labels, and a truncated linear model is adopted:

$$E_j(d_q,d) = \min(\lambda_{bp}, |d_q - d|), \quad (8)$$

where $\lambda_{bp} = 6.0$ is a constant controlling when the cost stops increasing. Equation 8 is defined under the assumption of piecewise-constant surfaces. The smoothness term is then the sum of the messages:

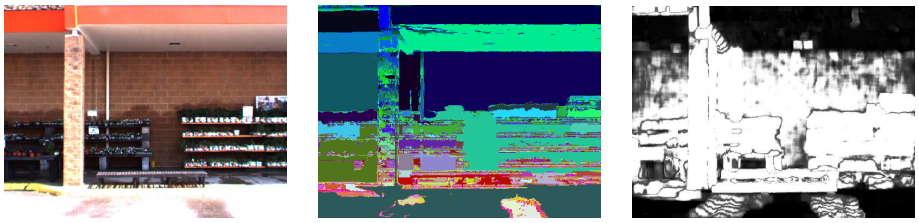
$$E^i_S(p,d) = \sum_{q\in N(p)} M^i_{q\to p}(d). \quad (9)$$

Rather than allow the global energy to converge, we stop after a certain number of iterations due to time constraints. Finally, the label d that minimizes $E(p,d)$ individually at each pixel is selected. A good example about how belief propagation corrects the errors introduced by the plane-fitting stereo is shown in Figure 2.

After BP refinement, we apply the color-weighted filter designed in Section 4.1 to E to help preserve the depth discontinuity under the assumption that color discontinuity is a strong indicator of depth discontinuity. Note that after BP refinement, the depth values of low confidence areas have been corrected, thus the confidence map should be updated too. Stereo fusion and another pass of plane-fitting refinement are then performed as in the plane-fitting stereo pipeline described in Section 4. Figure 3 shows depth maps produced by different stereo pipelines for visual comparison. Note that the BP-based plane-fitting stereo correctly captures the weakly-textured regions while preserving thin structures.

6 Results

For visual comparison, we ran the three stereo pipelines on an urban dataset. Figure 4 shows the depth maps produced by each method on a representative image, while Figure



(a) Reference image. (b) Color segmentation. (c) Stereo confidence.

(d) Window-based depth. (e) Plane-fitted depth. (f) BP-based depth.

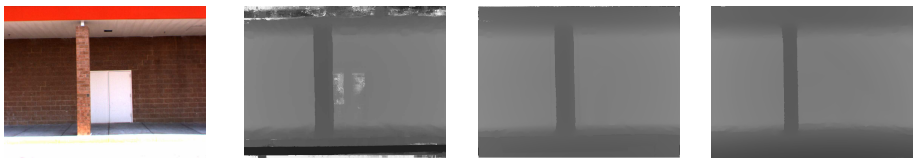
Figure 2: Due to strong illumination, part of the column in (a) is joined with the ground in color segmentation as seen in (b). In this case, the plane-fitted stereo (e) will fail. However, after BP refinement (f), the errors that appear in (e) are removed. The initial stereo confidence C_s and depth map are given in (c) and (d) respectively.

(a) Reference image. (b) Window-based. (c) Plane-fitting. (d) BP-based.

Figure 3: Visual comparison of depth maps. (b) is the depth map produced by a real-time local window-based stereo pipeline [1]. Plane fitting (c) corrects larger errors, such as the incorrect depth values in the textureless ground region. Belief propagation (d) refines the depth estimates locally and serves to preserve thin structures, such as those on the shopping cart.

5 shows their respective 3D models. The models are generated only with the highly-confident pixels in the depth map. If the confidence of a region is low, it will leave a hole in the 3D model. The two proposed stereo pipelines are capable of estimating depths planes for the weakly-textured areas where the window-based stereo clearly fails, such as the door in Figure 4(a). In Figure 5, the textureless areas on the ground result in a lack of confident depth estimates. However, the two proposed stereo pipelines successfully fill in most of these areas correctly. While plane fitting refines depth maps on a global scale, the effects of belief propagation are more local. These effects are primarily the smoothing of small errors and reduction of the fattening caused by the aggregation windows.

The two proposed stereo pipelines outperform the window-based stereo, while still providing good performance. With our settings the window-based stereo pipeline can



(a) Reference image (b) Window-based (c) Plane-fitting (d) BP-based

Figure 4: Visual comparison of representative depth maps associated with the 3D models in Figure 5.

Figure 5: top: Window-based stereo. middle: Plane-fitting stereo. bottom: BP-based plane-fitting stereo. Notice how we obtain good surfaces for the textureless regions where the window-based stereo fails.

process video data of resolution 512×384 and 48 depth hypotheses at about 18 frames per second using an NVIDIA Geforce 8800 GTX graphics card and an Intel Xeon 3.2GHz CPU, the Plane-fitting Stereo pipeline runs at about 8 frames per second, and the BP-based Plane-fitting Stereo pipeline runs at about 1 frame per second. Overall, the plane-fitting stereo pipeline achieves the best balance due to its fast processing time and ability to produce accurate reconstructions.

Our approach is primarily intended for urban and man made scenery with large textureless regions, rather than general stereo pairs; thus, the Middlebury datasets [14] do not exactly address the problem we are trying to solve. Nonetheless, we evaluated our BP-modified stereo and provide the results in Table 1. The values are the percentage of pixels with incorrect disparities on different image regions, along with their current rank. We omit the corresponding depth maps due to space constraints.

	Avg.	Tsukuba			Venus			Teddy			Cones		
	Rank	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
PlanefitBP	12.7	0.97$_{7}$	1.83$_{16}$	5.26$_{7}$	0.17$_{8}$	0.51$_{10}$	1.71$_{4}$	6.65$_{11}$	12.1$_{15}$	14.7$_{9}$	4.17$_{22}$	10.7$_{22}$	10.6$_{21}$

Table 1: Evaluation results on the Middlebury datasets with error threshold 1.

7 Conclusion and Future Work

In this paper, we focus on providing a fast, accurate solution to the reconstruction of weakly-textured regions that are common in urban environments. Our solution gives local smoothness to weakly-textured segments while preserving depth details in textured areas.

We do not currently consider any smoothness cost across the neighboring segments. Although we provide a solution with the BP-based modification, it is time consuming. We can reformulate this plane-fitting problem as an energy minimization problem which includes both data and smoothness terms. The data term associated with a 3D plane hypothesis will be the sum of all the euclidean distances from the 3D points to the plane, and the smoothness term will be a function measuring the similarity of the plane hypothesis in the current segment and the plane hypotheses in all its neighboring segments. Some stereo algorithms [12] are very adept at solving this energy minimization problem. These methods are far from being real-time because they use mean-shift color segmentation. However, this restriction is not an issue using our real-time segmentation method.

References

[1] A. Akbarzadeh, J.-M. Frahm, P. Mordohai, B. Clipp, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, H. Towles, D. Nistér, and M. Pollefeys. Towards urban 3d reconstruction from video. In *3DPVT*, 2006.

[2] C. Baillard and A. Zisserman. Automatic reconstruction of piecewise planar models from multiple views. *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, 02:2559, 1999.

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

[4] M.Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(8):993–1008, 2003.

[5] R.T. Collins. A space-sweep approach to true multi-image matching. In *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, pages 358–363, 1996.

[6] N. Cornelis, K. Cornelis, and L. J. V. Gool. Fast compact city modeling for navigation pre-visualization. In *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, pages 1339–1344, 2006.

[7] Edison. Edge detection and image segmentation system. http://www.caip.rutgers.edu/riul/research/code/EDISON/.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Intl. J. Comp. Vision*, 59(2):167–181, 2004.

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *Intl. J. Comp. Vision*, 70(1):41–54, 2006.

[10] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[11] S. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multiview stereo. In *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, 2001.

[12] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Proc. Intl. Conf. Pattern Recognition*, pages 15–18, 2006.

[13] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J-M. Frahm, R. Yang, D. Nister, and M. Pollefeys. Real-time visibility-based fusion of depth maps. In *Proc. Intl. Conf. Comp. Vision*, 2007.

[14] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Intl. J. Comp. Vision*, 47(1-3):7–42, 2002.

[15] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, pages 519–528, 2006.

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

[17] T. Werner and A. Zisserman. New techniques for automated architecture reconstruction from photographs. In *Proc. European Conf. Comp. Vision*, volume 2, pages 541–555, 2002.

[18] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. In *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, pages 2347–2354, 2006.

[19] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nistér. Real-time global stereo matching using hierarchical belief propagation. In *Proc. British Machine Vision Conf.*, pages 989–998, 2006.

[20] Q. Yang, R. Yang, J. Davis, and D. Nistér. Spatial-depth super resolution for range images. In *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, Minneapolis, MN, USA, 2007. IEEE Computer Society.

[21] K.-J. Yoon and I.-S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(4):650– 656, 2006.