Disparity Map using Suboptimal Cost with Dynamic Programming

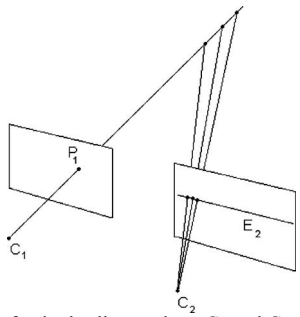
Eman F. Sawires[1], Alaa M. Hamdy[1], Fathy Z. Amer[1], E.M. Bakr[2]
[1]Electronics, Communication and Computer Department
Helwan University
1 Sherif st., Helwan
[2]Mechanical Engineering Department
Helwan University
1 Sherif st., Helwan

Abstract
1D optimization methods based on dynamic programming (DP) stereo are of practical interest because it can reconstruct an observed 3D optical surface very quickly and thus has potential for real-time applications. While being efficient, its performance is far from the state of the art because the vertical consistency between the scanlines is not enforced. 1D optimization based on dynamic programming for stereo correspondence is re-examined by applying it to the vertical consistency between the scanlines as opposed to the individual scanlines. To do this, a pixel is allowed to have a disparity with possibly sub-optimal cost for it in two directions.
Thus, the proposed algorithm is a truly global optimization method because disparity estimate at one pixel depends on the disparity estimates at all the other pixels, unlike the scanline based methods. Proposed algorithm is evaluated on the benchmark Middlebury database. The algorithm is very simple, so the proposed algorithm should be a good candidate for real time implementation. The results are considerably better than that of the scanline based methods. While the results are not the state of the art, the proposed algorithm offers a good trade off in terms of accuracy and computational efficiency.

Keywords
Stereo vision, Disparity, Dynamic programming, Scanlines, Suboptimal cost.

INTRODUCTION

Stereo vision is an imaging technique that can provide full field of view 3D measurements in an unstructured and dynamic environment. The foundation of stereo vision is similar to 3D perception in human vision.
Stereo vision technology is used in a variety of applications, including people tracking, mobile robotics navigation [1, 2], and mining. It is also used in industrial automation and 3D machine vision applications to perform tasks such as bin picking, volume measurement, automotive part measurement and 3D object location and identification.
The foundation of stereo vision is based on triangulation of rays from multiple viewpoints. Each pixel in a digital camera image collects light that reaches the camera along a 3D ray.
The most common method for extracting depth information from intensity images is by means of a pair of synchronized camera-signals, acquired by a stereo rig. The point-by-point matching between the two images from the stereo setup derives the depth images, or the so called disparity maps. This matching can be done as a one-dimensional search if accurately rectified stereo pairs in which horizontal scan lines reside on the same epipolar line are assumed [3], as shown in Fig. 1:



"Fig.1." Geometry of epipolar lines, where C_1 and C_2 are the left and right camera lens centers, respectively. Point P_1 in one image plane may have arisen from any of points in the line C_1P_1, and may appear in the alternate image plane at any point on the epipolar line E_2.

A point P_1 in one image plane may have arisen from any of points in the line C_1P_1, and may appear in the alternate image plane at any point on the so-called epipolar line E_2. Thus, the search is theoretically reduced within a scan line, since corresponding pair points reside on the same epipolar line. The difference on the horizontal coordinates of these points is the disparity. The disparity map consists of all disparity values of the image.
Detecting conjugate pairs in stereo images is a challenging research problem known as the correspondence problem, i.e. to find for each point in the left image, the corresponding point in the right one [4].
To determine these two points from a conjugate pair, it is necessary to measure the similarity of the points. The point to be matched should be distinctly different from its surrounding pixels. Thus, in the first stage of stereo matching, suitable features should be extracted. Several algorithms have been proposed in order to address this problem.
This paper is organized as follows. Section 2 presents stereo vision algorithms. Section 3 presents proposed algorithm. Section 4 presents experimental results. Section 5 presents the conclusion and the future work is presented in section 6.

2. Stereo Vision Algorithms

Stereo vision algorithms can be roughly divided into feature based [5, 6] and area based [7] algorithms. Feature based

algorithms use characteristics in the images like edges or corners to solve the correspondence problem in the two images.

The displacement between those features is used to build the disparity map and its density is directly related to the number of found features. Area based algorithms match blocks of pixels to find correspondences in the images. In the ideal case, each pixel can be found in the corresponding image as long as the search for the correct match keeps it within the image borders.

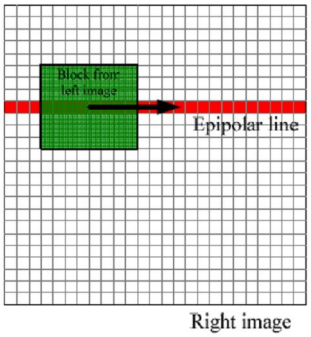
In general, matching algorithms can be classified into local and global methods. Local approaches utilize the color or intensity values within a finite window to determine the disparity for each pixel, usually make implicit smoothness assumptions by aggregating support [4]. Global approaches incorporate explicit smoothness assumptions and determine all disparities simultaneously by applying energy minimization techniques [8, 9].

2.1 Local Algorithms (Window-Based Algorithms)

There are some simple standard algorithms by using block matching and matching criteria, such as sum of absolute differences (SAD), sum of square differences (SSD) and cross correlation (CC) [10].

The blocks are usually defined on an epipolar line for matching ease. Each block from the left image is matched with a block in the right image by shifting the left block over the searching area of pixels in the right image, as shown in Fig. 2.

At each shift, the sum of comparing parameter e.g. intensity or color of the two blocks is computed and saved. The sum parameter is called “match strength”. The shift which gives a best result of the matching criteria is considered as the best match or correspondence.



"Fig .2." The block matching algorithm, computing each point of the left image block for every position through the corresponding epipolar line in the right image.

The following equations are often implemented for block matching algorithm:

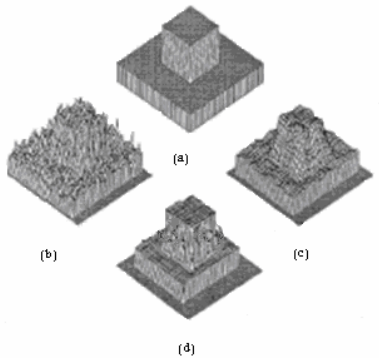
$$SAD(i,j,d) = \sum_{u=-w}^{w} \sum_{v=-w}^{w} |I_l(i+u,j+v) - I_r(i+u,j+v-d)| \quad (1)$$

$$SSD(i,j,d) = \sum_{u=-w}^{w} \sum_{v=-w}^{w} (I_l(i+u,j+v) - I_r(i+u,j+v-d))^2 \quad (2)$$

where I_l and I_r denote the left and right image pixel grayscale values, d is the disparity range, w is the window size and i, j are the coordinates (rows, columns) of the center pixel of the window for which the SAD and SSD are computed. At the disparity range, (d_{min} up to d_{max}), where the SAD and SSD are minimum for a pixel, this value is given as the corresponding pixel value for disparity map:

$$D(i,j) = \underset{d \in |d_{min}, d_{max}|}{\operatorname{argmin}} SAD(i,j,d) \quad (3)$$

The problem associated with this window-based approach is that the size of the correlation windows must be carefully chosen, as shown in Fig. 3. If the correlation windows are too small, the intensity variation in the windows will not be distinctive enough, and many false matches may result (small windows produce very noisy results, especially for low textured), if they are too large, resolution is lost, since neighboring image regions with different disparities will be combined in the measurement (fails to preserve image edges and fine detail), as will be discussed in section 4.



"Fig.3." The relation between edges and noise for different window sizes. (a) input – ground truth, (b) window size 3x3 (too noisy), (c) window size 7x7 (edges are blurred), (d) dynamic window (sharp edges and less noisy).

To solve the above problem and to increase the reliable disparities, it is necessary to have enough texture in the region to ensure a good match. This can be achieved by using a dynamic window size [11].

2.1.1 Local algorithms based on dynamic window size

To choose the appropriate window size that contains enough texture, the first step consists of calculating the local variation of image windows for the reference (left) image. Local variation (LV) is calculated according to the following formula [4].

$$LV(i,j) = \sum_{u=-w}^{w} \sum_{v=-w}^{w} |I_l(i+u,j+v) - \mu| \quad (4)$$

Where μ is the average grayscale value of image window;

Where the local variation for a given window central pixel is calculated according to the neighboring pixel grayscale values. The w is the selected square window size. Hence, first

the local variation over a window of 3x3 pixels is calculated and points with smaller local variation than a certain threshold value are marked for further processing. The local variation over a 5x5 range is computed for the marked points and is then compared to a threshold; Windows presenting smaller variation than threshold are marked for 7x7 range and so on.

Thus, LV is calculated by equation (4) and if LV is less than threshold, then window size is increased by two pixels.

The window size for each point of the image was set according to the results from the previous step. In order to find the corresponding points in the stereo pair images, a block matching method based on computing the SAD or SSD is calculated by equation (1) and equation (2) respectively.

The results of local algorithms are very far from the state of the art although dynamic window size is used, as will be discussed in section 4.

2.2 Global optimization

In global optimization, the constraints on the disparity map d are formulated into an objective function E (d) which is then minimized over all image pixels. A typical objective function has the following form [10]:

$$E(d) = E_{data} + \lambda E_{smooth}(d) \qquad (5)$$

The data term, $E_{data}(d)$, measures how well the disparity function d agrees with the input image pair. Using the disparity space formulation,

$$E_{data} = \sum_{(i,j)} C(i,j,d(i,j)) \qquad (6)$$

where C is the (initial or aggregated) matching cost . The smoothness term $E_{smooth}(d)$ encodes the smoothness assumptions made by the algorithm.

The global optimization methods will be broken in two groups: the 1D optimization methods and the 2D optimization methods.

The 1D optimization methods [10, 12] can be seen as drastically simplifying the objective function in equation (5). They enforce piecewise smoothness only in the horizontal direction, and so the optimization is reduced to one dimension. The $E_{smooth}(d)$ does not contain any terms based on neighboring pixels in the vertical direction.

Assuming that there are n scanlines, the energy function in equation (5) can be written as a sum of n energy functions, one for each scanline, and each one can be optimized separately from the others. This optimization can be performed efficiently and exactly using dynamic programming, this approaches work by computing the minimum-cost path through the matrix of all pairwise matching costs between two corresponding scanlines [13].

The 1D optimization methods are not truly global optimization methods because the disparity estimate at a pixel depends only on the disparity estimate of pixels on the same scanline, but is completely independent of the disparity estimates on the other scanlines.

The performance of 1D optimization methods is far from the state of the art, as shown in section 4, since piecewise smoothness is enforced only in the horizontal direction. The most noticeable artifact which distinguishes the resulting disparity maps of such methods is the horizontal "streaking" which results from the lack of coherence in the vertical direction. Most methods [14, 15, 16] try to improve results by post processing between the scanlines, with various degrees of success. The advantage of the 1D optimization methods is that they are simple to implement and are efficient.

The 2D optimization methods enforce piecewise smoothness in both horizontal and vertical directions. Traditional 2D optimization approaches include simulated annealing [17], continuation methods [18], and mean-field annealing [19]. Though interesting from the theoretical point of view, these methods are rather inefficient.

Recently, graph-cuts [20, 21] and belief propagation methods [22, 23, 24] have been applied quite successfully to optimize equation (5). These methods are relatively efficient and produce excellent results according to the recent stereo evaluation on data with ground truth conducted by [10]. Still these methods are far from real time, and their theoretical complexity is not quite clear because they are iterative.

3. Proposed algorithm

The motivation behind the proposed work is to use the powerful and efficient optimization tool provided by 1D dynamic programming, and apply it to a structure more suited to enforce piecewise continuity than a scanline.

The estimate of disparity at one pixel depends on the estimate of disparity at all the other pixels. Thus the proposed dynamic programming is a truly semi global optimization algorithm.

The proposed algorithm is not a 1D optimization method because it operates across both vertical and horizontal dimensions.

To implement dynamic programming efficiently, the methods used are developed by [25, 26]. Typically, if an image has j columns and number of possible disparity values is h, then the straightforward dynamic programming takes O (jh^2) time, but the running time can be reduced to O (jh) which is the complexity of our method. Modeling is started on objective function of the type in equation (5). The data term in equation (6) can now be written as:

$$E_{data}(d) = m(d_{(i,j)}) \qquad (7)$$

Let i, j be the coordinates of the pixel in the left image and $d_{(i,j)}$ be the value of disparity map d at pixel . Let m ($d_{(i,j)}$) be the matching penalty for assigning disparity $d_{(i,j)}$ to pixel , in our frame work $m(d_{(i,j)})$ can be sum of absolute differences (SAD) to compare the image regions,

$$m(d_{(i,j)}) = SAD(i,j,d) = \sum_{u=-3}^{3} \sum_{v=-3}^{3} |I_l(i+u,j+v) - I_r(i+u,j+v-d)| \qquad (8)$$

This means that for every pixel in the left image, the 3-by-3-pixel block is extracted around it and then a search is

performed along the same row in the right image for the block that best matches it, where $d \in h$.

Let $s(d_n, d_{n\pm k})$ be the smoothness penalty for assigning disparities d_n and $d_{n\pm k}$, where $n \in h$ and k is integer number from 0 to 6, In particular, each disparity is constrained to lie with ±6 values of its neighbors' disparities.

In the proposed framework, $s(d_n, d_{n\pm k})$ can be

$$s(d_n, d_{n\pm k}) = | d_{n\pm k} - d_n | \quad (9)$$

Thus the energy function required to be optimized is given by equation (5).

$$E(d_{(i,j)}) = m(d_{(i,j)}) + \lambda\, s(d_n, d_{n\pm k}) \quad (10)$$

Then the minimum value of the energy in equation (10) can be written as:

$$E(d_{(i,j)}) = \min(m(d_{(i,j)}) + \lambda\, s(d_n, d_{n\pm k})) \quad (11)$$

The optimal disparity assignment for $d_{(i,j)}$ can be written as:

$$I(d_{(i,j)}) = \operatorname{argmin}(m(d_{(i,j)}) + \lambda\, s(d_n, d_{n\pm k})) \quad (12)$$

The proposed algorithm operates across both vertical and horizontal dimensions, so the previous equations are:

$$E(d_{(i,j)}) = \min(m(d_{(i,j)}) + \lambda\, s(d_n, d_{n\pm k}) + E(d_{(i,j-1)}) + \gamma E(d_{(i-1,j)})) \quad (13)$$

$$I(d_{(i,j)}) = \operatorname{argmin}(m(d_{(i,j)}) + \lambda\, s(d_n, d_{n\pm k}) + E(d_{(i,j-1)}) + \gamma E(d_{(i-1,j)})) \quad (14)$$

For any other row i , i>1 and other column j , j>1

Where λ is the disparity penalty, γ is the coefficient of suboptimal cost in horizontal direction.

The following must be noted:

For the first row and first column $E(d_{(i,j-1)})$ and $E(d_{(i-1,j)})$ equal zero.

For the first row, and any other column, j>1, $E(d_{(i-1,j)})$ equals zero.

For any other row i , i>1 and first column, $E(d_{(i,j-1)})$ equals zero.

The output of the previous stage is used and dynamic programming is implemented to calculate the disparity map.

The proposed algorithm is simple to describe and implement, but also much more efficient, as will be presented in section 4. As expected, the results fall in the middle range. The state of the art results are given by the more computationally costly 2D optimization algorithms. Results of the proposed algorithm are far better than those of methods based on 1D optimization.

4. Experimental Results

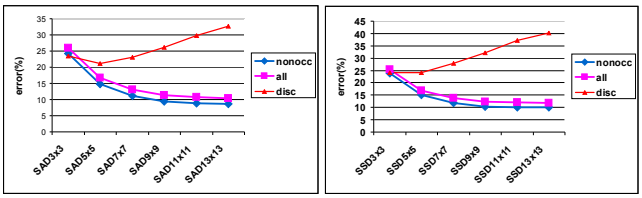
The algorithms have been tested on the benchmark Middlebury Database [27], there are four stereo pairs, named Tsukuba, Venus, Teddy and Cones, respectively. The error is computed as the percentage of pixels is far from the true disparity by more than one. These statistics are collected for all non-occluded regions (shown in column nonocc), for all (including half-occluded) regions (shown in column all), and finally regions near depth discontinuities (shown in column disc).

The Middlebury database has two scenes which have much more complex geometry, called teddy and cones.

Consider in detail the "Teddy" stereo pair [27]: it has a large disparity range (0 to 59) and generally more complex surface geometry than other scenes using benchmarking. The ground truth has quarter-pixel accuracy. The "Teddy" scene presents three main challenges for stereo reconstruction – complicated geometry (e.g. green teddy and plants), large nontextured slant planar surfaces (e.g. the background and roof of the toy house) and it has large occluded areas, e.g. the background area behind the top of the roof.

Also the "cones" stereo pair: it has a large disparity range (0 to 59) and generally more complex surface geometry, the ground truth has quarter-pixel accuracy.

SAD and SSD had been implemented; the results have been tested on the benchmark Middlebury database, Fig. 4 shows nonocc and all will be improved if the window size is increased but at expense of disc, as expected.



(a) (b)

"Fig .4." The relation between nonocc , all and disc for different window sizes (a) SAD, (b) SSD for Tsukuba .

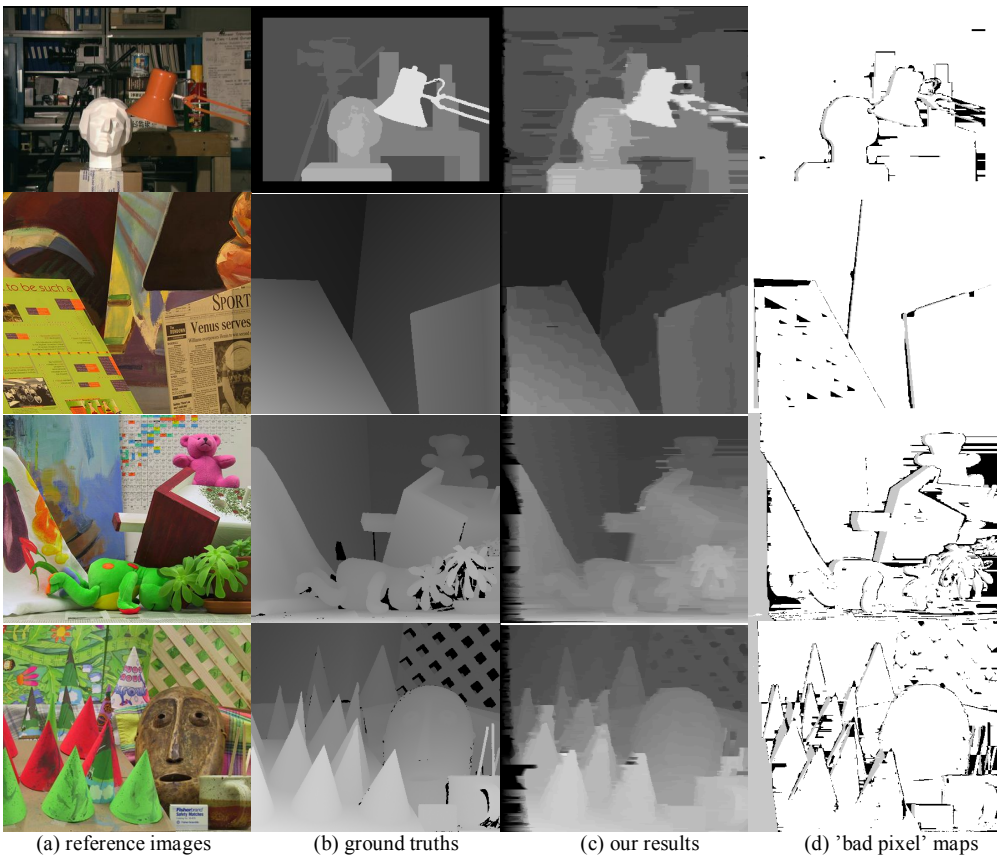
SAD based on dynamic window (SADDW) and SSD based on dynamic window (SSDDW) have been implemented; the results have been tested on the benchmark Middlebury database. Table 1 shows the results far from the middle range. This means the dynamic window did not improve the results. So, DP is of practical interest, because the DP can handle lack of texture and occlusion in the image.

The proposed algorithm has been tested on the benchmark Middlebury database [27]. The disparity maps for the proposed algorithm are in Fig. 5.

Table 2 shows the accuracy of the proposed algorithm at the best parameters used. As expected, the results fall in the middle range.

This direct comparison of proposed algorithm to the 1D optimization methods may not be fair. Each of the 1D optimization methods uses a different cost function and different types of post processing to alleviate the "horizontal streaking" artifacts. So, 1D scanline optimization has been implemented with proposed objective function, but has optimized it on the scanlines. Table 3 shows the accuracy of 1D optimization algorithm.

A comparison is done between the result of the proposed algorithm and the result of 1D algorithm. The results noted in the proposed algorithm are better than those of 1D optimization algorithm. This is because the average percent of bad pixels is decreased by 3.3% and the second reason is that the proposed algorithm results on Teddy pair (Teddy pair is so difficult as mentioned before) are better than many algorithms[27].



(a) reference images (b) ground truths (c) our results (d) 'bad pixel' maps

"Fig.5." Results using the Middlebury datasets: Tsukuba, Venus, Teddy and Cones. Pixels with a disparity error greater than one Pixel are displayed in the 'bad pixel' maps.

"Table 1." SADDW,SSDDW result on Middlebury database, T=threshold.

Algorithms	Tsukuba Nonocc all disc T=40			Venus Nonocc all disc T=20			Teddy Nonocc all disc T=50			Cones Nonocc all disc T=100			Average percentage of bad pixels
SADDW	12.4	14.4	25.4	8.8	10.3	33.4	24	31.1	40.5	17.5	26.5	33.5	23.3
SSDDW	13.1	15.1	28.8	8.9	10.4	36.1	23.7	30.3	43.2	14.6	24.1	34.2	23.5

"Table 2." The results of proposed algorithm on Middlebury database at best parameters.

Image pair	Tsukuba Nonocc all disc			Venus Nonocc all disc			Teddy Nonocc all disc			Cones Nonocc all disc			Average percentage of bad pixels
Best parameters	λ 0.1		γ 0.2	λ 0.9		γ 0.9	λ 0.5		γ 0.3	λ 0.7		γ 0.6	
Proposed Algorithm	4.31	6.44	13.9	2.71	4.36	19.7	9.61	19	22	6.71	17	17.9	12.0

"Table 3." The results of 1D optimization algorithm on Middlebury database at best parameter.

Image pair	Tsukuba Nonocc all disc			Venus Nonocc all disc			Teddy Nonocc all disc			Cones Nonocc all disc			Average percentage of bad pixels
1D optimization	4.47	6.58	13.9	8.64	10.2	34	10.4	19.6	22.6	9.84	19.8	23.2	15.3

5. Conclusion

This research presents a suboptimal cost with dynamic programming algorithm that is a global optimization method to calculate disparity map. The disparity for each pixel depends on the disparity for all others pixels. The proposed algorithm offers a good trade off in terms of accuracy and computational efficiency.

6. Future work

In the future, the proposed algorithm will be implemented by FPGA to decrease the processing time to be more compatible with real time applications.

ACKNOWLEDGMENTS

Many thanks to the authors of [10] for providing the test images and the unique evaluation test bed.

REFERENCES

[1] D. Murray, C. Jennings, "Stereo vision based mapping for a mobile robot.", in: Proceedings of the IEEE International Conference on Robottics and Automation, May 1997.
[2] D. Murray, J.J. Little, "Using real-time stereo vision for mobile robot navigation.", Autonom. Robots, 8 (2): 161–171, April 2000.
[3] Teerapat Chinapirom, U.W., and Ulrich Rückert, "Steroscopic Camera for Autonomous Mini-Robots Applied in KheperaSot League.", System and Circuit Technology, Heinz Nixdorf Institute, University of Paderborn, 2001.
[4] Georgoulas, C., Kotoulas,L., Sirakoulis,G., Andreadis, I., Gasteratos, A., "Real-Time Disparity Map Computation Module.", Journal of Microprocessors & Microsystems, 32(3):159–170, 2008.
[5] Hajar Sadeghi, Payman Moallem, and S. Amirhassn Monadjemi, "Feature Based Dense Stereo Matching using Dynamic Programming and Color.", 2008.
[6] A. Baumberg, "Reliable feature matching across widely separated views.", CVPR, vol.1:774–81, 2000.
[7] L. Di Stefano, M. Marchionni, S. Mattoccia, "A fast area-based stereo matching algorithm.", Image Vis Comput, 22 (12): 983–1005 ,2004.
[8]Q. Yang, L. Wang, and N. Ahuja, "A constant-space belief propagation algorithm for stereo matching.", CVPR, 80-87, 2010.
[9]T. Montserrat, J. Civit, O. Escoda, and J.-L. Landabaso, "Depth estimation based on multiview matching with depth/color segmentation and memory efficient belief propagation.", ICIP,2009.
[10] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.", IJCV, 47(1-3):7–42, April 2002.
[11] A.J.Lipton ,"Local Application of Optic Flow to Analyse Rigid versus Non-Rigid Motion." , ICCV Workshop on Frame-Rate Vision, 1999.
[12] I. Cox, S. Hingorani, S. Rao, and B. Maggs. "A maximum likelihood stereo algorithm. Computer Vision.", Graphics and Image Processing, 63(3):542–567, 1996.
[13] Hajar Sadeghi, Payman Moallem, and S. Amirhassn Monadjemi, "Feature Based Dense Stereo Matching using Dynamic Programming and Color.", International Journal of Computational Intelligence 4;3 (2008).
[14] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search.", TPAMI, 2:449–470, 1985.
[15] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to- pixel stereo.", IJCV, 35(3):1–25, December 1999.
[16] A. Bobick and S. Intille, "Large occlusion stereo.", IJCV, Vol.3:181-200, September 1999.
[17] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images.", IEEE Transactions on Pattern Analysis and Machine Intelligence, 6:721–741, 1984.
[18] A. Blake and A. Zisserman, "Visual Reconstruction.", MIT Press, 1987.
[19] D. Geiger and F. Girosi, "Parallel and deterministic algorithms from MRF's: Surface reconstruction.", IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(5):401–412, May 1991.
[20] V. Kolmogorov and R. Zabih, "Multi-camera scene reconstruction via graph cuts.", In ECCV02, page III: 82 ff., 2002.
[21] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts.", IEEE Trans. Pattern Analysis Machine Intell., Vol. 23:11, pp. 1222–1239, 2001.
[22] A. Klaus, M. Sormann and K. Karner, "Segmentbased stereo matching using belief propagation and a self-adapting dissimilarity measure.", ICPR, Vol. 3: 15–18, 2006.
[23] J. Sun, N. Zheng, and H. Shum, "Stereo matching using belief propagation.", PAMI, 25(7):787–800, July 2003.
[24] P. Felzenszwalb and D. Huttenlocher, "Efficient belief propagation for early vision.", In CVPR04, pages I: 261–268, 2004.
[25] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition.", IJCV, 61(1):55–79, January 2005.
[26] Park, CS; Park, HW, "A robust stereo disparity estimation using adaptive window search and dynamic programming search.", PR(34), No.12, PP.2573-2576, December 2001.
[27] http://vision.middlebury.edu/stereo/