Fast Stereo Matching by Iterated Dynamic Programming and Quadtree Subregioning

Carlos Leung[1], Ben Appleton[1], Changming Sun[2]
[1]Electromagnetics and Imaging, ITEE,
The University of Queensland, Brisbane, QLD 4072, Australia
[cleung, appleton]@itee.uq.edu.au
[2]CSIRO Mathematical and Information Sciences,
Locked Bag 17, North Ryde, NSW 1670, Australia
changming.sun@csiro.au

Abstract

The application of energy minimisation methods for stereo matching has been demonstrated to produce high quality disparity maps. However the majority of these methods are known to be computationally expensive, requiring minutes or even hours of computation. We propose a fast minimisation scheme that produces strongly competitive results for significantly reduced computation, requiring only a few seconds of computation. In this paper, we present our iterated dynamic programming algorithm along with a quadtree subregioning process for fast stereo matching.

1 Introduction

The study of computational stereo has undergone intensive research since its inception in the 1970s. Research is now focussed on three main issues: occlusion modelling, global optimisation techniques and real-time implementations [4]. Increasing commercial interest in real-time 3D reconstruction requires the development of fast algorithms that can produce high quality disparity maps. Current hardware stereo systems employ 'greedy' algorithms, which assign each pixel the disparity giving the best local match. Reconstruction based on simple local correspondence methods can be improved by incorporating global constraints and structural information into the stereo matching process.

One such class of global correspondence methods are those based on dynamic programming (DP). However since DP is typically applied to independent scanlines, methods that employ this technique suffer from interscanline inconsistencies. Several studies have addressed this issue by applying postprocessing to iteratively improve the reconstruction, enforcing interscanline constraints. These techniques include minimising the number of horizontal and vertical discontinuities [5], estimating vertical slopes [1], and using edge maps [7]. Whilst these heuristics improve interscanline consistencies they do not entirely solve the problem. The framework proposed in this paper does not exhibit dimensional bias and thus is immune to the interscanline problem.

Another class of global correspondence techniques are those that formulate the stereo problem into a two-dimensional energy minimisation framework. By designing an energy functional whose minima will correspond to good stereo reconstructions, the aim is to compute a disparity function that minimises this energy. Sun [10] proposed a two-stage dynamic programming technique to compute disparity surfaces of maximum total correlation. In recent years algorithms based on graph cuts and iterated graph cuts have been proposed to solve the optimisation problem [2, 6, 8]. Graph cut methods produce excellent results at the cost of orders of magnitude greater computation than dynamic programming techniques.

In this paper we present two new techniques for fast stereo matching. We propose an iterated dynamic programming (IDP) algorithm which minimises an energy function that incorporates both intrascanline and interscanline regularity. This framework overcomes the interscanline inconsistency problem inherent to DP techniques and produces comparable results to existing energy minimisation algorithms. We also propose a quadtree subregioning (QSR) algorithm for the fast computation of the matching cost volume in a multiscale framework. Timings and results are presented to demonstrate the quality and speed of the proposed techniques.

2 Energy Function

The goal of dense two-view reconstruction is to recover the depth of each pixel from a stereo image pair. The stereo pair is assumed to be rectified such that corresponding horizontal scanlines lie in the same epipolar plane. A *disparity function* $d(x,y)$ represents the horizontal displacement for each point of the reference image and is related to the depth of that point in the scene. To each disparity function d we associate an energy $E[d]$ quantifying the matching quality. We minimise a discontinuity-preserving energy function which may be viewed as a Bayesian labelling of a first-order Markov random field. This energy function includes terms for data fidelity and regularisation:

$$E[d] = \sum_{(x,y)} c(x,y,d(x,y)) + \sum_{(x_1,y_1)\sim(x_2,y_2)} e(|d(x_1,y_1) - d(x_2,y_2)|). \qquad (1)$$

The first term accounts for the matching cost c of pixel correspondences. Many matching metrics have been proposed in the literature [4]. In this paper we consider the zero-mean normalised cross-correlation (ZNCC) and the sum of absolute differences (SAD) metrics.

Stereo reconstruction based solely on matching criteria is an ill-posed problem which has many solutions. The second term of Eq. (1) imposes the assumption of regularity onto the disparity function to obtain solutions which are considered likely from prior knowledge, where the edge function $e(\cdot)$ is selected to penalise discontinuities in d and $\sim$ is the neighbourhood relation between points. A variety of edge functions have been proposed, including quadratic functions, discontinuity-preserving functions, and terms dependent on intensity differences [9].

Although discontinuity-preserving energy functions are desirable in stereo reconstruction, Boykov et al. showed that the minimisation of such an energy is NP-hard by analogy to the Potts model [3]. They proposed a multiway graph cut framework which can compute a strong local minimum of such energy functions. While their optimisation scheme relies on an iterated application of minimum cuts, we propose a fast alternative which optimises the same energy function using iterated dynamic programming.

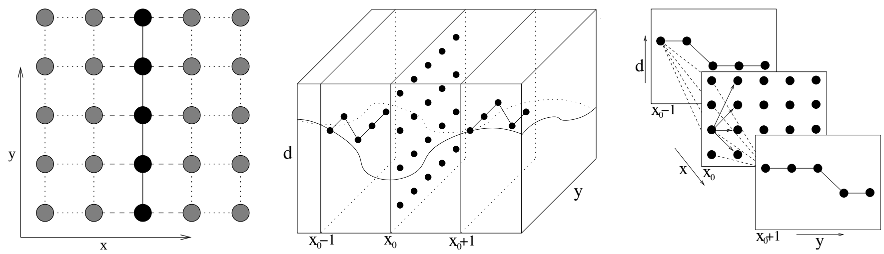

Figure 1: (left) An instance of a column $d(x_0,\cdot)$ of the disparity function to be optimised. Solid lines denote *active* edges, dashed lines denote *passive* edges. Dotted lines are not considered by the current optimisation. (middle) The corresponding $(x_0,\cdot,\cdot)$ plane in the cost volume to be optimised during the iteration process. Depicted are the neighbouring planes whose disparity values affect the current optimisation. (right) The planar trellis on which dynamic programming is performed. Arrows denote the *active* edges while dashed lines denote the *passive* edges.

3 Fast Stereo Matching

3.1 Iterated dynamic programming

In contrast to graph cuts which can compute the optimal binary-labelling of two-dimensional energy functions, dynamic programming can compute the optimal multi-labelling of a one-dimensional energy function. Our energy minimisation takes advantage of this, optimally relabelling entire rows and columns of the disparity function at once.

IDP iteratively optimises a disparity function $d(x,y)$ by relabelling the pixels along a single row or column. While doing so, it leaves the remainder of the disparity function untouched. In this way we may reduce a multi-dimensional energy minimisation problem to a sequence of one-dimensional subproblems. Each of these one-dimensional subproblems may be solved to optimality using dynamic programming. As the algorithm proceeds the energy decreases monotonically, converging when no row or column remains which may be relabelled to further reduce the energy. We therefore compute a strong minimum of the energy function — at convergence the replacement of any horizontal or vertical line cannot decrease the energy.

An example of the operation of IDP is depicted in Fig. 1. In the vertical case, fixing $x = x_0$, the energy function Eq. (1) becomes:

$$E[d(x_0,\cdot)] = \kappa + \sum_y e(|d(x_0,y) - d(x_0,y+1)|) + \sum_y (c(x_0,y,d) + e(|d(x_0,y) - d(x_0 \pm 1,y)|)) \quad (2)$$

Here we neglect the image borders for brevity. The first term κ absorbs components of the energy function which are unrelated to the current line being optimised. The second term accounts for the energy of the *active* edges, which are the interactions between neighbouring points along the vertical line $d(x_0,\cdot)$. The third term depends on the values of $d(x_0,\cdot)$ at individual points along the current line, absorbing the matching costs and *passive* edges. Observe that we have converted the minimisation problem posed by Eq. (1) into a one-dimensional subproblem whose global minimum may be efficiently

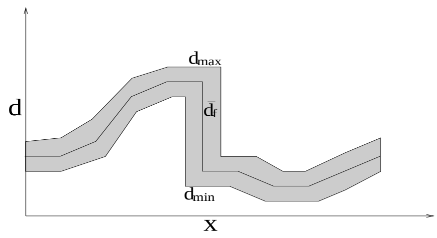

Figure 2: Given the coarse scale estimate $\bar{d}_f$, all computations may be restricted to the narrowband between $d_{\min}$ and $d_{\max}$ (shaded).

computed using DP. The algorithm monotonically reduces the energy of the disparity estimate and converges when every horizontal and vertical line is optimal. Here are the steps of our IDP algorithm:

Iterated Dynamic Programming

1. Compute the matching cost volume
2. Initialise $d(x,y)$ to zero
3. For each row and column in sequence:
   (a) Form Eq. (2) along the current row or column
   (b) Solve Eq. (2) using dynamic programming
4. If convergence has not been reached, return to step 3

An advantage of our energy minimisation scheme is that it allows for arbitrary edge functions. While the multiway cut framework of Kolmogorov and Zabih [6] requires that the smoothness term be a metric or semi-metric, iterated dynamic programming has no constraints on the choice of edge penalty. Empirically, we have found that the following discontinuity-preserving edge function produces good results:

$$e(\Delta d) = \begin{cases} 0 & |\Delta d| = 0 \\ K_1 & |\Delta d| = 1 \\ K_2 & |\Delta d| \geq 2 \end{cases} \tag{3}$$

Here K_1 and K_2 are regularisation parameters $(K_1 \leq K_2)$. As we are dealing with integer disparities Δd will also be an integer. Note that this edge function is a semi-metric.

3.2 Coarse-to-fine scheme

The computational cost of working at the full disparity range across the entire image is prohibitively high. Instead we take a coarse-to-fine approach, using a coarse estimate of the disparity surface to restrict the range of disparities that need to be considered to a narrow band.

Let $d_c(x,y)$ be the disparity surface computed at the coarse scale and $d_f(x,y)$ the disparity function to be computed at the next finer scale. From $d_c(x,y)$ we may derive bounds $d_{\min}(x,y) \leq d_f(x,y) \leq d_{\max}(x,y)$ on the range of disparities to consider in the computation of d_f. Here we use power of two downsampling, such that the fine scale has twice the resolution of the coarse scale. Then the coarse scale estimate localises spatial discontinuities in the disparity function at half of the resolution of the full scale. Likewise, the disparity values themselves are twice as heavily quantised in the coarse scale. The restricted search range is depicted in Fig. 2. We obtain these bounds $d_{\min}(x,y)$ and $d_{\max}(x,y)$ by the following procedure:

Disparity range estimation

1. Upsample d_c to the fine resolution and scale by a factor of 2 to obtain $\overline{d_f}$
2. Set $d_{\min} = \overline{d_f} - 1$, $d_{\max} = \overline{d_f} + 1$
3. Erode $d_{\min}$ and dilate $d_{\max}$ by 1 pixel

These bounds greatly reduce the work required to compute the matching costs and the disparity range within which the iterated dynamic programming algorithm must search.

3.3 Quadtree subregioning

In order to perform fast stereo matching, the matching cost c needs to be efficiently computed. Sun [10] applied a recursive cross correlation scheme to evaluate the matching costs inside a rectangular box of size (X,Y) with D disparities in $O(XYD)$ computation. Unfortunately this cannot be easily extended to arbitrary shapes, and therefore the application of this scheme to compute a narrow band of values is non-trivial. Sun proposed a rectangular subregioning scheme to compute a set of boxes which includes the narrow band. Rectangular subregioning proceeds by dividing the image into horizontal stripes, merging neighbouring stripes of similar disparity ranges. This procedure is also carried out in the vertical direction on each resulting stripe. The disadvantage of this method is that the merging of horizontal stripes must operate over the full width of the image and therefore may not adapt well to scenes with a wide range of disparities in the horizontal direction.

By merging rectangles rather than stripes, a more adaptive partitioning may be obtained. Our aim is to obtain large rectangles with small disparity ranges and small rectangles with large disparity ranges, so that the overall cost of computing the matching costs within R rectangles, $\sum_{i=0}^{R-1}(X_i Y_i D_i)$ may be reduced. We propose a subregioning algorithm which takes a divide-and-conquer approach to merging rectangles, obtaining an optimal subregioning.

Given a box in which a narrow band of matching costs must be evaluated, we may either compute the full box or subdivide this box into quarters. If we know the minimal computation of these quarters, it is simple to evaluate whether merging these may reduce the computation cost. Recursively applying this splitting produces the optimal QSR. Beginning with the input image, the algorithm proceeds as follows:

Quadtree Subregioning

1. Given a box B of dimensions (X_i, Y_i), partition in x and y to give four children B_{mn} where $m,n \in \{1,2\}$
2. Compute the minimal computation cost $C(B)$:
   (a) Compute the *merge* cost for window size (x_w, y_w)
   $C_{\text{merge}}(B) = (X_i + x_w)(Y_i + y_w)D_i + C_{\text{overhead}}$

(b) Compute the *split* cost $C_{\text{split}}(B) = \sum_{m,n} C(B_{mn})$

(c) Determine the minimal computation cost $C(B) = \min(C_{\text{merge}}(B), C_{\text{split}}(B))$

(d) Label the box as *split* or *merge* accordingly

3. Beginning with the largest box B, extract quadtree subregions

   (a) If this box is labelled *merge*, add box to list

   (b) If this box is labelled *split*, recurse to children B_{mn}

4 Experimental Results

In this section we present experimental results demonstrating the high quality of the stereo reconstructions computed using IDP and QSR. We compare our new algorithm with existing methods and with ground truth on a variety of real images. All experiments have been performed on a 1.8GHz Pentium IV under the Windows operating system. The algorithms have been implemented in C++ and compiled with standard optimisation flags.

Reconstruction Quality

A recent survey of stereo correspondence algorithms has been presented by Scharstein and Szeliski [9]. All major stereo correspondence algorithms were compared on four stereo data sets with associated ground truth. Here we present the results of IDP on the same data set. Fig. 3 demonstrates our reconstructions.

Error measures were also defined to quantitatively describe the quality of the computed correspondences. The error metric used by the survey to evaluate the overall quality of the reconstructions was the percentage of mis-matched pixels in unoccluded regions. Here we compare the results of our method to competing algorithms using the same error metric. Table 1 summarises the error percentages using a fixed set of parameters across all images, as well as using parameters optimised for each image. In the fixed parameter experiment we used the SAD metric with 3×3 comparison windows. The edge function parameters were $K_1 = 200$ and $K_2 = 1000$.

In order to compare the quality of iterated dynamic programming to the state of the art, we have included the method of Kolmogorov and Zabih [6] which is known to produce excellent results. This method uses the α-expansion graph cut method to minimise an energy function similar to that given in Eq. (1) with additional terms modelling occlusions. As the performance of Kolmogorov and Zabih's method is only given for fixed parameters, we have also included the results for an alternative graph cut method, α-β swap moves, proposed by Boykov et al. [3] implemented in [9]. Included in the table also are performance measures for two other methods considered in the survey paper: a dynamic programming algorithm which considers occlusions, and a scanline optimisation algorithm which solves the same energy functional as the one described in the paper, except that vertical smoothness terms are ignored.

From Table 1, we observe that the results produced by IDP are competitive with those computed using graph cuts. Although our method uses dynamic programming, it does not suffer from the classic problem of interscanline inconsistency. Included in the table also are performance measures for a dynamic programming algorithm which considers occlusions, and a scanline optimisation algorithm which solves the same energy functional as the one described in the paper, except that vertical smoothness terms are ignored [9].

The use of a multi-directional smoothing energy omits the need to perform additional post-processing, common to scanline optimisation algorithms. Our relatively high error percentage in the Tsukuba data set is due to the small disparity range in the scene. Our energy minimisation scheme computes the optimal multi-labelling of a row or column, and generally performs better on scenes with large disparity ranges. By contrast graph cut methods prefer images with few disparities, converging to the optimum on images with two disparities. This difference is evident in the results on the Venus and Map images with higher disparity ranges where IDP gives the strongest performance.

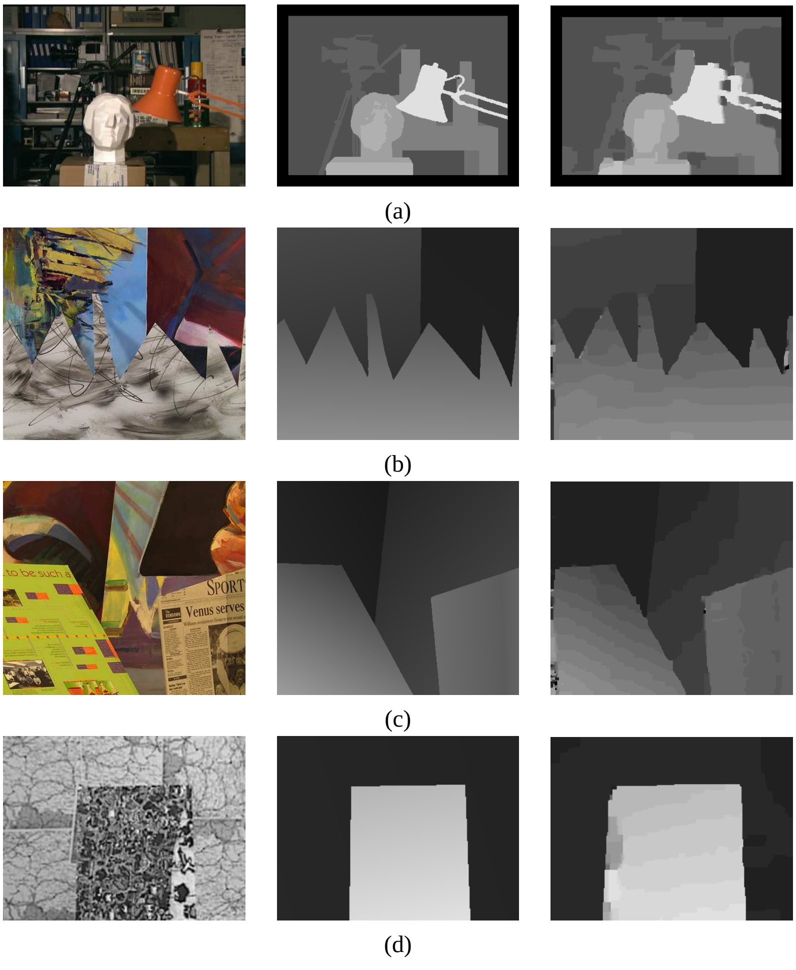

Figure 3: The results of IDP on the data set from Scharstein and Szeliski's stereo survey [9]. The middle column depicts the ground truth for each image. Images on the right are stereo reconstructions computed on: (a) **Tsukuba** data set 384×288 computed using 10 disparity levels (0.83s). (b) 434×383 **Sawtooth** image set using 21 disparity levels (3.89s). (c) 434×380 **Venus** data set using 21 disparity levels (4.24s). (d) **Map** image set 284×216 computed using 31 disparity levels (1.07s).

Table 1: A comparison of the reconstruction quality of IDP against four other algorithms from the survey: Kolmogorov and Zabih (GC+occl.) [6], Scharstein and Szeliski (GC, SO, DP) [9]. Quoted are the percentage of mis-matching pixels in unoccluded regions, using both fixed and optimised parameters for each image.

	Tsukuba		**Sawtooth**		**Venus**		**Map**	
	fixed	best	fixed	best	fixed	best	fixed	best
GC+occl.	1.27	—	0.36	—	2.79	—	1.79	—
GC	1.94	1.94	1.30	0.98	1.79	1.48	0.31	0.09
SO	5.08	4.66	4.06	3.47	9.44	8.31	1.84	1.04
DP	4.12	3.82	4.84	3.70	10.10	9.13	3.33	1.21
IDP	3.27	2.94	1.83	1.01	1.52	1.34	0.17	0.11

Table 2: The computation times for each component of the algorithm. Observe that QSR significantly reduces the computational load of the matching cost computation. The combination of IDP and QSR produces a very efficient algorithm for stereo correspondence.

Image	**Parking Meter** (512×480×31)		**Pentagon** (512×512×21)		**Fruit** (512×512×46)	
	No QSR	QSR	No QSR	QSR	No QSR	QSR
Matching Cost: Points ($\times 10^6$)	5.7	1.7	4.1	1.0	11.8	3.0
Time (s)	1.97	1.00	2.47	1.43	4.49	2.50
Optimisation Time (s)	0.81	0.81	1.44	1.44	1.86	1.86
QSR Computation Time (s)	—	0.05	—	0.07	—	0.05
Total Time (s)	2.78	1.86	3.91	2.94	6.35	4.41

Running Times

While methods based on dynamic programming typically require only a few seconds of computation, graph cut methods require 10-30 minutes [9]. IDP retains the computational benefits of a dynamic programming formulation while producing results competitive with graph cut approaches. Here we present the running times for each component of our algorithm demonstrating its fast computation and the effectiveness of QSR (Table 2).

Quadtree subregioning was introduced to minimise the amount of computation required when computing the matching cost volume in a coarse-to-fine approach. Fig. 4 demonstrates examples of QSR applied to three real images. Observe that regions of constant disparity tend to form large partitions while regions of highly varying disparity such as object boundaries tend to have smaller boxes. QSR reduces the number of metric evaluations from XYD to $\sum_{i=0}^{R-1}(X_i Y_i D_i)$ for R subregions. Table 2 lists the total number of metric evaluations in a coarse-to-fine matching with and without the use of QSR. We note that QSR reduces the number of metric evaluations has been reduced by a factor of four. The time required to compute the matching costs has been approximately halved on all images, clearly outweighing the small overhead of the QSR algorithm.

For the four data sets of Fig. 3, the time required was 0.83 seconds for the Tsukuba data set, 3.89 seconds for Sawtooth, 4.24 seconds for Venus and 1.07 seconds for Map. The quality and speed of our reconstructions establishes IDP as a competitive energy minimisation scheme for stereo matching.

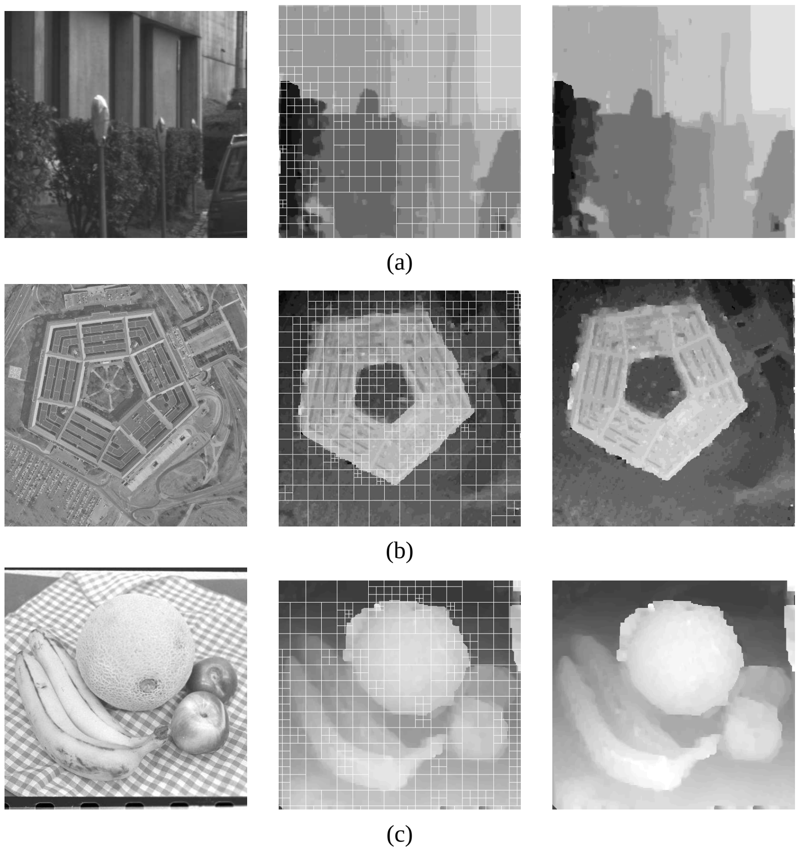

Figure 4: The results of quadtree subregioning on three real images: (a) **Parking meter**. (b) **Pentagon**. (c) **Fruit**. Depicted are: (middle) the boxes overlaid on the computed disparity function and (right) the final disparity function.

5 Conclusions

We have presented a new algorithm, iterated dynamic programming, for fast stereo reconstruction. Taking advantage of dynamic programming's ability to compute the optimal multi-labelling of a one-dimensional energy function, we proposed an iterated dynamic programming scheme that can minimise a discontinuity-preserving energy function. We have also presented an algorithm for the fast computation of the cost volume by computing an optimal quadtree subregioning of the disparity image.

Results have been presented and compared to existing stereo energy minimisation algorithms. These results demonstrate that iterated dynamic programming is strongly competitive with graph cut techniques while maintaining the computational advantages of dynamic programming techniques. Combined with quadtree subregioning, iterated dynamic programming computes high quality reconstructions in seconds rather than minutes.

References

[1] P.N. Belhumeur. A Bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3):237–260, August 1996.

[2] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *ICCV*, pages 489–495, 1999.

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

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

[5] I. Cox, S. Hingorani, S. Rao, and B. Maggs. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, May 1996.

[6] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV*, volume 3, pages 82–96, 2002.

[7] Y. Ohta and T. Kanade. Stereo by intra- and inter- scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139–154, March 1985.

[8] S. Roy and I. J. Cox. A maximum-flow formulation of the N-camera correspondence problem. In *ICCV*, pages 492–499, Bombay, India, January 1998.

[9] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dence two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):7–42, April–June 2002.

[10] Changming Sun. Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques. *International Journal of Computer Vision*, 47(1/2/3):99–117, April–June 2002.