

# Fast Algorithms for Stereo Matching and Motion Estimation\*

Changming Sun

CSIRO Mathematical and Information Sciences, Locked Bag 17, North Ryde, NSW 1670, Australia  
changming.sun@csiro.au

**Abstract**—This paper presents fast algorithms for similarity measure, stereo matching, panoramic stereo matching, and image motion estimation. The disparity map for the stereo images is found in the 3D correlation coefficient volume by obtaining the global 3D maximum-surface by using our two-stage dynamic programming (TSDP) technique. Fast panoramic stereo matching is carried out using a cylindrical maximum surface technique. Optical flow or image motion estimation is obtained using 3D shortest path techniques. A variety of synthetic and real images have been tested, and good results have been obtained.

**Keywords:** Rectangular subregioning (RSR), 3D Maximum-Surface, Two-stage dynamic programming (TSDP), Fast panoramic stereo matching, Circular shortest path, Cylindrical maximum surface, Motion estimation, Optical flow, 3D shortest path.

## I. INTRODUCTION

The correspondence problem in stereo vision and image motion concerns the matching of points or other kinds of primitives such as edges and regions in two or more images such that the matched image points are the projections of the same point in the scene. The disparity map or motion field obtained from the matching stage may then be used to compute the 3D positions of the scene points given the imaging geometry. Matching techniques can be divided broadly into area-based and feature-based image matching, or a combination of them. Other types of stereo matching methods such as pixel-based [4], diffusion-based [5], wavelet-based [6], phase-based [7], and filter-based [8] have also been developed. There are also several types of methods for estimating image motion or optical flow [9].

In this paper we will present techniques for fast stereo matching, fast panoramic stereo matching and image motion estimation. The rest of the paper is organised as follows: Section II proposes our method for fast calculation of similarity measure. Section III presents our method for fast stereo matching by finding the maximum-surface in the 3D correlation volume using the TSDP technique. Section IV presents algorithms for fast panoramic stereo matching. Section V gives our method for fast motion estimation. Section VI discusses the reliability and computation speed issues of our algorithms. Section VII gives concluding remarks.

## II. FAST SIMILARITY MEASURES

Similarity or dissimilarity is the guiding principle for solving the stereo matching or motion correspondence problem. Different similarity measures have been used in the literature for matching, and their performance and computation costs vary [10], [11]. The most commonly used similarity measure is the cross correlation coefficient especially for stereo matching. But direct calculation of ZNCC is computationally expensive. Faugeras *et al* [12] developed a recursive technique to calculate the correlation coefficients which are invariant to the correlation window size. Sun [13], [14] used box-filtering technique for fast cross correlation. The following subsections describe methods for achieving fast correlation on the whole image and our technique of using rectangular subregioning for fast similarity measure calculation.

\* This paper is based on the following three published or about to be published articles: [1], [2], [3].

### A. Fast Cross-Correlation on the Whole Images: Review

To achieve fast cross-correlation calculation, one needs to have fast ways to obtain the mean and variance of a window and cross covariance values of two local windows in two input images. Fast calculation of local mean and variances can be achieved using the box-filtering technique. Cross-covariance of two images can be obtained using only a few multiplications by exploiting techniques similar to the fast calculation for the mean and variance by using box-filtering.

The correlation of two windows in the two images is performed along the same horizontal scan line. For any point in the left image, if the search window is assumed to be within  $[-w, +w]$  in the right image, then the value of disparity  $d$  varies from  $-w$  to  $+w$ . In an algorithm for fast correlation calculation, one first fixes on one particular  $d$  for all the points in the left image and calculates the local cross correlation between the whole left image and the whole shifted right image of the amount  $d$  using box-filtering technique. After this, for every point on the left image we have a local cross correlation value for the shift of  $d$ . Then we increase the number of  $d$  by 1, and repeat the process of correlation calculation until the value of  $d$  has gone through  $[-w, +w]$ . For each  $d$ , a plane of correlation coefficients is produced. Putting all of these planes together we have a 3D correlation volume. The size of the volume depends upon the image row and column numbers  $M, N$  and the maximum disparity search range  $D(=2w+1)$  as shown in Figure 1. The complexity of the algorithm is  $O(MND)$ . The storage space needed for the correlation coefficients is in the order of  $4MND$  bytes if float data type is used.

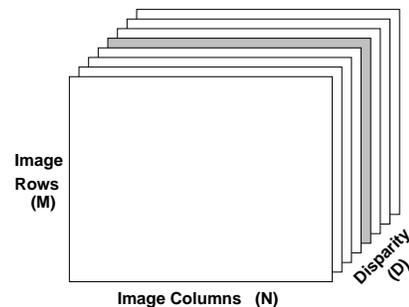


Fig. 1. An illustration of the 3D correlation coefficient volume obtained after using the fast correlation method. The grey plane in the middle of the volume corresponds to the coefficients when  $d = 0$ . The size of the volume is  $MND$ .

### B. Rectangular Subregioning (RSR)

Rather than work with the whole image during the fast image correlation stage as described in the previous subsection, we could work with subimages to speed up the correlation calculation further. If the input image is divided into  $R$  subimages or rectangular subregions, the computation complexity will be  $\sum_{i=0}^{R-1} (M_i N_i D_i)$ , where  $M_i, N_i$  are the row and column numbers for the  $i$ th subimage or region, and  $D_i$  is the disparity search range over this subimage. We call the process of segmenting the input images into rectangular

subimages as rectangular subregioning (RSR). Because the disparity search range  $D_i$  is obtained in a much smaller region ( $M_i N_i$ ),  $D_i$  is expected to be smaller than  $D$ , which is obtained for the whole image. Even when it is not much smaller, the size of this region ( $M_i N_i$ ) is much smaller than the input image. It is anticipated that  $\sum_{i=0}^{R-1} (M_i N_i D_i)$  will be smaller than  $MND$ , especially when the disparity changes a lot within the whole image.

### C. Fast Similarity Measure in 2D Search

It is possible to compute the cross correlation using only a few multiplications by exploiting techniques similar to those described in Section II-A with search in both x- and y- directions. For each pair of  $d_x$  and  $d_y$ , a plane of correlation coefficients is produced.  $d_x$  and  $d_y$  can be varied over  $[-w_x, +w_x]$  and  $[-w_y, +w_y]$  to produce a correlation volume of size  $MND_x D_y$ . The complexity of our algorithm is  $O(MND_x D_y)$ . It is independent of the local window sizes. The storage space needed for the correlation coefficients is in the order of  $MND_x D_y$  floating points. If SAD or SSD is used, the data type could be integer or short integer rather than floating points.

## III. STEREO MATCHING USING MAXIMUM-SURFACE

From the previous section, we have obtained a 3D cross correlation coefficient volume as shown in Figure 1 using fast cross correlation working on rectangular subregions for stereo images.

### A. Maximum-Surface in the Volume

In this section, we will approach the issue of obtaining disparity map from the 3D correlation coefficient volume using dynamic programming techniques, which is computationally efficient. We developed a method to obtain a maximum-surface from a 3D volume using a two-stage dynamic programming (TSDP) technique. This maximum-surface cuts through the 3D volume from the top to the bottom as illustrated in Figure 2.

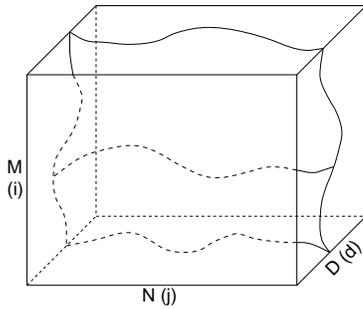


Fig. 2. The illustration of the 3D maximum-surface which give the maximum accumulation of values in the 3D cross correlation coefficient volume.

Now we describe our algorithm for the maximum-surface extraction in a 3D volume of size  $MND$  using our fast TSDP method. The first stage of the algorithm is to obtain an accumulated intermediate 3D volume in the vertical direction for each vertical  $j$  slice. Assume  $C(i, j, d)$  is the correlation coefficient value in the input 3D volume at position  $(i, j, d)$ , where  $0 \leq i < M$ ,  $0 \leq j < N$ , and  $0 \leq d < D$ . We create an intermediate array  $Y(i, j, d)$  which contains the accumulated values of the maximum cross correlation coefficients for each vertical  $j$  slice of the same 3D volume using dynamic programming techniques say from top to bottom, i.e. when  $i$  changes from 0 to  $M-1$ . For those values in the top horizontal slice of the volume, i.e. when  $i = 0$ ,

$$Y(0, j, d) = C(0, j, d) \quad (1)$$

i.e. the top (horizontal) slice of  $Y$  is a copy of the top slice of  $C$ . For the remaining horizontal slices of the volume, the  $Y$  values at each position is obtained by using the following recursion which is a typical dynamic programming formula:

$$Y(i, j, d) = C(i, j, d) + \max_{t:|t| \leq p} Y(i-1, j, d+t) \quad (2)$$

where  $p$  determines the number of local values that need to be checked.

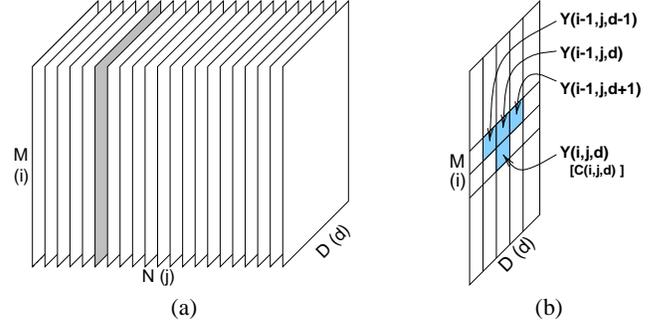


Fig. 3. Obtaining the  $Y(i, j, d)$  volume. (a) shows the 3D volume  $Y$  with a vertical slice in grey; (b) illustrates the positions of the  $Y$  values at each iteration.

After the recursion of the first stage dynamic programming described in the previous paragraph, we now move to a second stage of the TSDP algorithm using volume  $Y$  to obtain the disparity map for the input stereo images. In this stage, we work in the horizontal direction. Starting from the bottom of the 3D volume  $Y$  where maximum values have been accumulated, we select the 2D horizontal slice with  $i = M-1$ , i.e. the bottom slice for disparity estimation. From this 2D matrix of size  $ND$ , a shortest-path from left to right is obtained using dynamic programming techniques. The sum of the values along this path gives the maximum value which is also the maximum summation value along the whole 3D surface. This obtained path is related to the disparities for the last or bottom row of the input image. The distance of each point along this path to the middle line is the obtained disparity for the same  $x$ -positioned point of the input image.

We then move from the bottom slice of  $Y$  upwards. When calculating the disparity for row number  $i-1$ , we use the result obtained for row number  $i$ . We now select the 2D horizontal slice number  $i-1$  of the 3D volume  $Y$ , and mask out those values which are more than  $p$  position away from the shortest-path obtained from row number  $i$ . Then a new shortest-path is obtained in this 2D matrix from left to right. This process of obtaining shortest-path is repeated until the shortest-path for the first row of the image is obtained.

Putting all the shortest-paths for each of the scan line together forms a 3D surface within the 3D volume of  $Y$ . Because successive shortest-path for each scan line is obtained in the neighbourhood of the previous path position, the 3D maximum-surface technique gives more consistent disparities. The complexity of the TSDP algorithm is linear with respect to the size of the 3D volume, i.e.  $O(MND)$ .

### B. Algorithm Steps

The steps of our algorithm, which uses the combination of RSR and TSDP, for fast stereo matching in a coarse-to-fine scheme are:

- 1) Build image pyramids with  $P$  levels (from 0 to  $P-1$ ), from the original left and right images.
- 2) Initialize the disparity map as zero for level  $k = P-1$  and start stereo matching at this level.

- 3) Perform stereo matching using the method described in Sections II-III which includes:
  - a) Segment images into rectangular subregions based on the current disparity map;
  - b) Perform fast zero mean normalised correlation to obtain the correlation coefficients for each subregions and build a 3D correlation coefficient volume for the whole image;
  - c) Use the two-stage dynamic programming technique to find the 3D maximum-surface, which will then give the disparity map as described in Section III.
- 4) If  $k \neq 0$ , propagate the disparity map to the next level in the pyramid using bilinear interpolation, set  $k = k - 1$  and then go back to Step 3; if  $k = 0$ , go to Step 5.
- 5) Display disparity map.

### C. Experimental Results

This section shows some of the results obtained using our RSR+TSDP algorithm described earlier. The input left and right images are assumed to be rectified epipolar images. Implementations of the Roy's [15], Cox's [16] and Sun's [13] methods are used for comparison. The codes for Roy's and Cox's methods are downloaded from their web pages.

Figures 4-5 show some results using real images. The input Pentagon images are shown in Figure 4(a,b). The matching results for our method, Roy's, Cox's and Sun'97 methods are given in Figure 4(c,d,e,f). The input Fruit scene images are shown in Figure 5(a,b). The matching results for our method, Roy's, Cox's and Sun'97 methods are given in Figure 5(c,d,e,f). From the results shown in Figures 4 and 5, it can be seen that our RSR+TSDP method gives more consistent results than the other three methods. Many other types of real images have also been tested, and good results have been obtained.

The typical running time for the algorithm on a  $256 \times 256$  image with about 30 pixels disparity is in the order of several hundred milliseconds with a 500MHz Pentium III running Linux. Table I gives some of the typical running times of the algorithm on different sizes of images with different disparities using whole image correlation and the RSR methods. The time shown for "User time1" is obtained without using the RSR method, while the time shown for "User time2" is obtained by using the RSR method. It can be seen that the time spent by the algorithm using RSR method is almost half of the time without using the RSR method.

Table II gives some of the typical running times of the 2D matrix and 3D maximum-surface algorithms on different size of images. The computation time for the 3D surface method is only slightly longer than that of the 2D path method. Table III shows the computation times for Roy, Cox and our algorithms on three pair of images. Roy's algorithm takes much longer to finish compared with other two algorithms. Our method is also much quicker than Cox's method.

TABLE III  
Running times of different algorithms.

Image name	Image size	Disp. range	Roy's method	Cox's method	Our method
RDS	$300 \times 300$	10	300.63s	1.44s	0.45s
pm	$512 \times 480$	25	374.83s	4.28s	1.75s
pentagon	$512 \times 512$	25	462.47s	5.43s	1.62s

### D. Fast Stereo Matching Web Demo

Our fast stereo matching algorithm web demo is at: <http://extra.cmis.csiro.au/IA/changs/stereo/>

## IV. FAST PANORAMIC STEREO MATCHING USING CYLINDRICAL MAXIMUM SURFACES<sup>1</sup>

Panoramic images are becoming increasingly popular in image-based virtual environment representations and digital photography. Panoramic stereo images are also becoming available for 3D applications. The panoramic stereo images that we are interested in are the 360-degree stereo images on a cylindrical surface. Panoramic stereo images can be obtained by a number of methods. They can be generated by mosaicking images from a rotating camera [18], [19], or by using special imaging optics [20], [21], [22], [23].

Some matching methods for panoramic stereo images use standard window based correlation search [21], [24], [19]. Because of the special wrap-around property for the panoramic stereo images, special care needs to be taken during the stereo matching process. Zheng and Tsuji used circular dynamic programming for matching vertical features in panoramic images [25]. Li *et al* used tensor voting techniques for matching multiperspective panoramas [26].

Here we propose to use cylindrical maximum surface techniques for carrying out dense panoramic stereo matching considering the continuity of neighboring epipolar scanlines. We can use the techniques described in Section II-A for fast correlation calculation. As a result of the fast similarity measure, we obtain a 3D correlation volume. The size of the volume depends upon the image row and column numbers  $M$ ,  $N$  and the maximum disparity search range  $D$  as shown in Fig. 6(a). As the inputs are panoramic stereo images, the 3D correlation coefficient volume obtained actually forms a cylindrical volume as shown in Fig. 6(b). We will find a maximum surface within this cylindrical volume for disparity estimation.

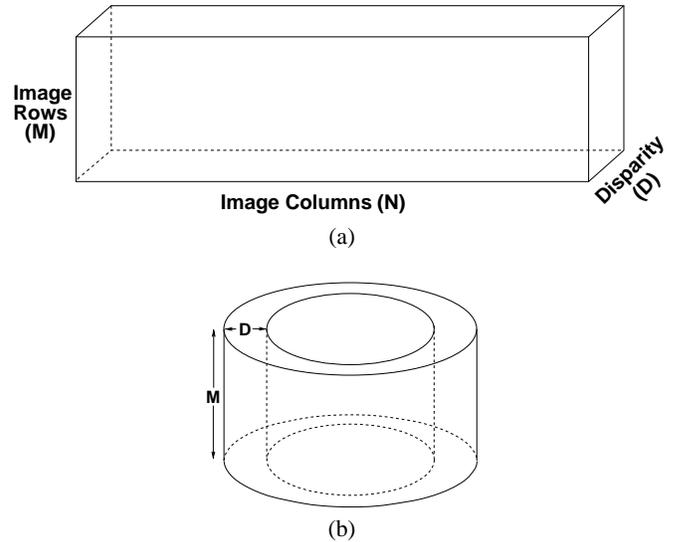


Fig. 6. An illustration of the 3D correlation coefficient volume for panoramic stereo images obtained after using the fast correlation method. (a) shows the 3D volume; and (b) shows the same volume in cylindrical shape.

### A. Circular Shortest Path in 2D Matrix

In panoramic stereo matching, a horizontal slice of the cylindrical volume as shown in Fig. 7 has the property that the left most and the right most columns are connected. This 2D slice can also be shown in the format of a cylindrical surface as in [27]. In traditional stereo matching, dynamic programming (DP) techniques have been used to obtain shortest paths to estimate disparities [28], [13], [29].

<sup>1</sup>This section is a joint work with Professor Shmuel Peleg at School of Computer Science and Engineering, The Hebrew University, Israel [2].

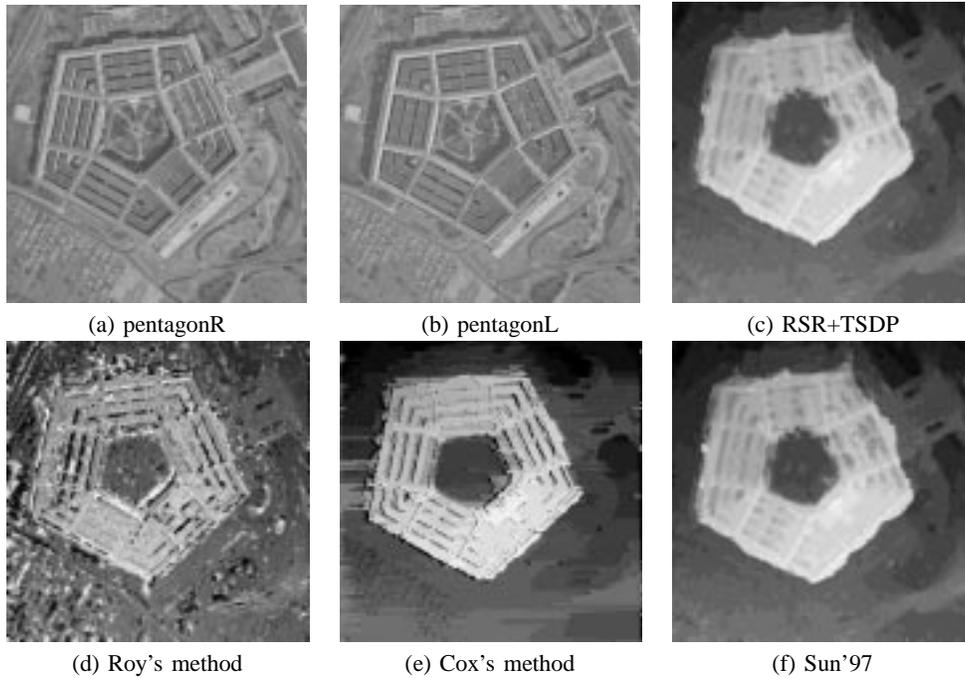


Fig. 4. *Pentagon stereo. (a) and (b) are the right and left input images. (c) Results obtained using our method (RSR+TSDP). (d) Results obtained using Roy's method. (e) Results obtained using Cox's method. (f) Results obtained using the method described in [13]. (Images (a,b) courtesy of Bill Hoff at the Univ. of Illinois [17]).*

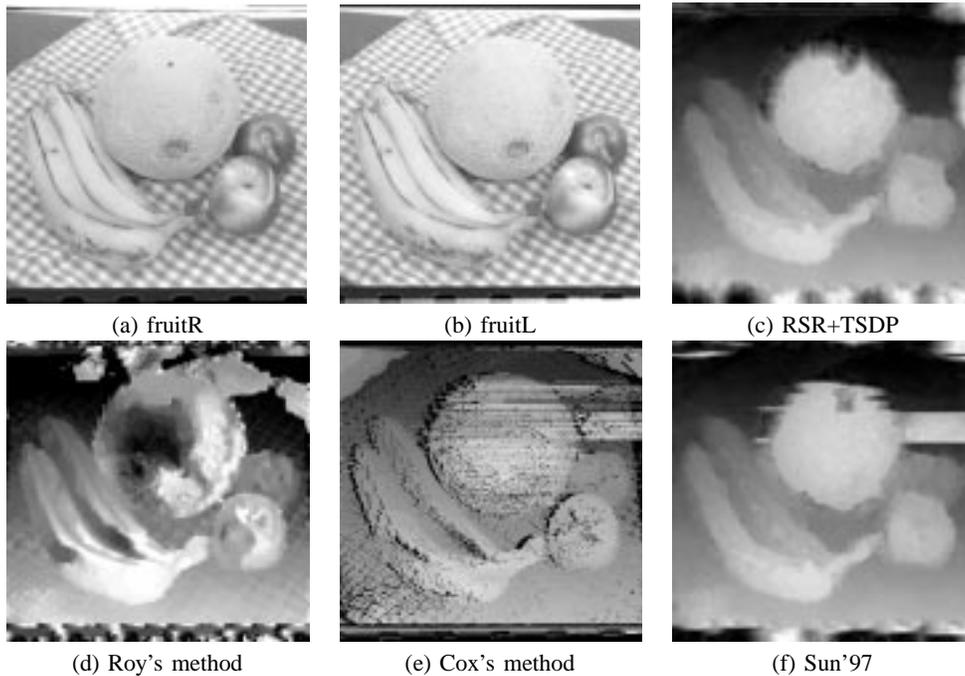


Fig. 5. *Fruit stereo. (a) and (b) are the right and left input images. (c) Results obtained using our method (RSR+TSDP). (d) Results obtained using Roy's method. (e) Results obtained using Cox's method. (f) Results obtained using the method described in [13]. (Images (a,b) courtesy of Bill Hoff at the University of Illinois [17]).*

TABLE I

Running times of the whole image correlation and the RSR algorithms on different images. The dynamic programming stage of this test runs on 2D matrix. The size of the correlation window is  $9 \times 9$ . The reduction ratio  $r$  used in the pyramid generation process is 2.

Image name	Image size	Pyramid levels	Search range	Disparity range	User time1	User time2
ball	256×256	3	[-4,4]	[-19,7]	0.53s	0.32s
pentagon	512×512	3	[-2,2]	[-10,10]	2.42s	1.39s
circuit	512×512	3	[-5,5]	[-21,23]	3.36s	1.59s
flat	1000×1000	4	[-3,3]	[-31,23]	16.86s	7.51s

TABLE II

Running times of our algorithms on different images. The size of the correlation window is  $9 \times 9$ . The reduction ratio  $r$  used in the pyramid generation process is 2. Both of these algorithms use RSR.

Image name	Image size	Pyramid levels	Search range	Disparity range	Method 2D path	Method 3D surface
ball	256×256	3	[-4,4]	[-19,7]	0.32s	0.37s
pentagon	512×512	3	[-2,2]	[-10,10]	1.39s	1.50s
circuit	512×512	3	[-5,5]	[-21,23]	1.59s	1.82s
flat	1000×1000	4	[-3,3]	[-31,23]	7.51s	7.53s

For panoramic stereo matching, we can use circular shortest path (CSP) extraction technique to obtain a CSP in each 2D correlation matrix (sized  $ND$  as shown in Fig. 7) so that the starting and ending positions of this path are connected.

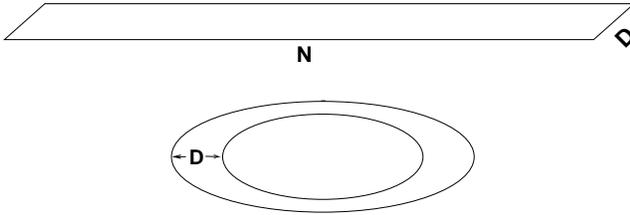


Fig. 7. One slice of the 3D cylindrical volume shown in Fig. 6.

Five algorithms (MSA: multiple search algorithm; IPA: image patching algorithm; MBTA: multiple backtracking algorithm; Combination algorithm of IPA and MBTA; and Approximate algorithm) in [27] and one (BBCSP: circular shortest path by branch and bound) in [30] have been developed for CSP extraction on regular grids or images when the left and the right columns of the grid are neighbours.

One can simply use the CSP extraction algorithm mentioned earlier to obtain a CSP for each slice of the 3D cylindrical volume independently for the disparity estimation of the panoramic stereo images. However this approach does not take information from neighbouring scanlines into account (apart from the windowing effect during correlation). In the following subsection, we use this CSP extraction technique to obtain a 3D surface in the cylindrical volume for panoramic stereo matching. We intend to obtain a maximum 3D cylindrical surface rather than a number of independent CSPs.

### B. Maximum Surface in a Cylindrical Volume

In this subsection, we will approach the issue of obtaining the panoramic stereo disparity map from the cylindrical shaped 3D correlation coefficient volume using a CSP technique, which is computationally efficient. A cylindrical maximum surface which cuts through the cylindrical volume from the top to the bottom as shown in Fig. 8 is obtained in two steps. The cylindrical maximum surface

gives the maximum summation of the correlation coefficients along the surface inside the cylindrical volume.

The process of obtaining such surface is similar to that for obtaining the 3D surface described in Section III except replacing shortest path with circular shortest path.

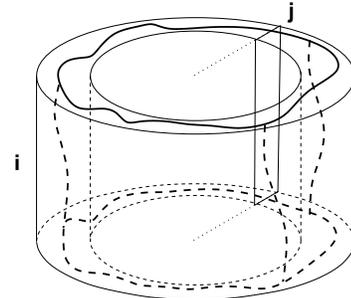


Fig. 8. The cylindrical maximum surface which give the maximum accumulation of cross correlation values in the cylindrical volume. The vertical rectangle in the figure shows one vertical slice at position  $j$  of the cylindrical volume.

Putting all the CSPs obtained for each of the scanline together form a 3D cylindrical surface within the 3D volume. Because successive CSP for each scanline is obtained in the neighbourhood of the previous path position, the cylindrical maximum surface gives more consistent disparities. The result of obtaining this cylindrical surface is that the summation of the correlation values on this surface is maximum.

### C. Experimental Results

This section shows some of the results obtained using our method described in previous sections. The input left and right panoramic stereo images are assumed to be rectified epipolar images.

Figure 9 shows the different results obtained for a portion of a panoramic stereo images by using just the CSP algorithm or the cylindrical maximum surface technique. Figure 9(a) is the result obtained using the CSP algorithm for each horizontal slice of the 3D correlation volume independently. Figure 9(b) gives the result

obtained using the cylindrical maximum surface technique. Note that there is a white streaking around the top left region and there is a dark streaking around the bottom right region in Fig. 9(a).

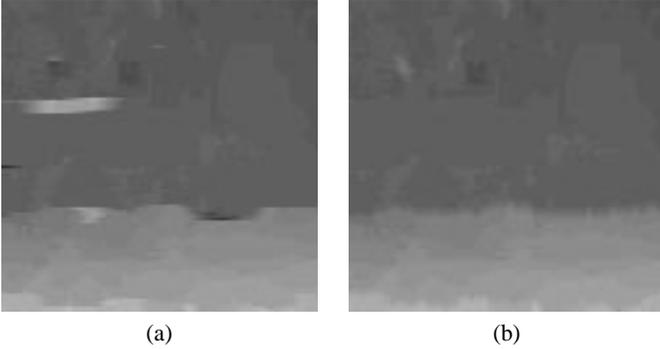


Fig. 9. Different disparity results obtained for a portion of the panoramic stereo images. (a) Disparity obtained using CSP for each horizontal slice of the 3D correlation volume; (b) Disparity obtained using the cylindrical maximum surface technique.

Figure 10 gives the results obtained by using our methods described in previous sections. The first two images are the left and the right input panoramic stereo images. The third image is the disparity map obtained. The running time for the algorithm on a  $1324 \times 120$  image is about 0.34 seconds on a 1.7GHz Linux PC. We are showing the disparity for every points of the stereo images. The work by Li *et al* [26] for depth estimation from multiperspective panoramas by the use of tensor voting techniques seems to give good results. Their algorithm, however, takes about 60 minutes on a Pentium III 550MHz.

## V. FAST MOTION ESTIMATION

Optical flow or image motion is the displacement of each image pixel in an image sequence. Image motion estimation is a fundamental issue in low-level vision and is used in many applications such as robot navigation, object tracking, image coding, and structure reconstruction. There are several types of methods for estimating image motion or optical flow [9]. These methods can be divided into correlation-based [31], [32], [33], [34], energy-based [35], phase-based [36], gradient-based [37], [38], [39], [40] methods, and orthogonal dynamic programming method [41]. In this section we develop algorithms for image motion estimation by using fast correlation and dynamic programming techniques.

### A. Correlation Volume

The result of the correlation calculation described in Section II-C is a volume containing the correlation coefficients as shown in Fig. 11. The size of the volume depends upon the image size  $MN$  and the motion search ranges  $D_x$  and  $D_y$ . Each pixel in the first image has  $D_x D_y$  correlation coefficients in the corresponding search region in the second image. These coefficients are stored in a 1D vector in the 3D volume as shown in Fig. 11. This vector represents the 2D search region shown in the right hand side of the same figure. There are  $N$  such 2D search regions containing the correlation coefficients in each horizontal scanline of the input image. These 2D regions can be stacked together to produce a 3D volume of correlation coefficients with dimensions  $D_x D_y N$  for each scanline of the image as shown in Fig. 12. This correlation volume will be used to obtain motion vectors for this scanline using 3D shortest path method to be described in Section V-B.

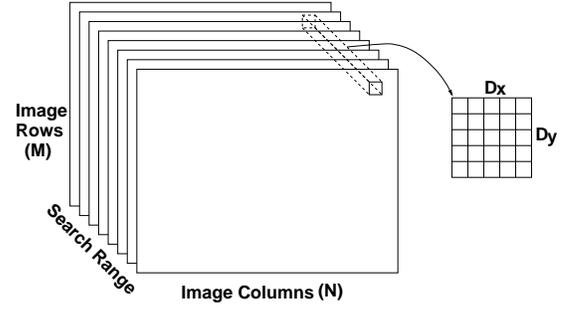


Fig. 11. An illustration of the correlation volume obtained after using our fast correlation method. The number of correlation planes equals the size of the search region  $D_x D_y$ .

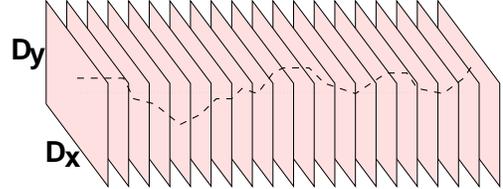


Fig. 12. Correlation volume for each scanline. Each plane in the volume contains the correlation coefficient values within a search region. There are  $N$  such planes for each scanline.

### B. Shortest Path in 3D Using Dynamic Programming

We propose a method which uses a 3D shortest path through the 3D correlation volume for each scanline of the input image to produce a consistent set of motion vectors. This volume is one horizontal slice of the correlation volume shown in Fig. 11. The position of the path indicates the best motion vector for this scanline. Because the path is continuous, the motion vectors obtained for neighbouring pixels are more consistent with each other.

Now we describe our algorithm for the shortest path extraction in a 3D volume using efficient dynamic programming techniques. For  $0 \leq p < D_x$ ,  $0 \leq q < D_y$  and  $0 \leq k < N$ , let  $C(p, q, k)$  be the cost (or the correlation coefficient value) of the  $(p, q, k)$ th value in the 3D volume of size  $D_x D_y N$ . Array  $Y(p, q, k)$  contains the accumulated values and  $K(p, q, k)$  has the position which produces the local maximum value. When  $k = 0$ ,

$$Y(p, q, 0) = C(p, q, 0) \quad (3)$$

i.e. the first plane of  $Y$  is a copy of the first plane of  $C$ . For the remaining planes ( $k$ th plane) of the volume, the  $Y$  values at each position is obtained using the following recursion:

$$Y(p, q, k) = C(p, q, k) + \max_{s, t: |s| \leq 1, |t| \leq 1} Y(p + s, q + t, k - 1) \quad (4)$$

The values of  $s, t$  which achieves the maximum in Eq. (4) during each iteration is stored in  $K$ .

$$K(p, q, k) = \operatorname{argmax}_{s, t: |s| \leq 1, |t| \leq 1} Y(p + s, q + t, k - 1) \quad (5)$$

The values stored in volume  $K$  are used to backtrack along the best path from the maximum value in the last plane of  $Y$ . After the  $Y$  and  $K$  volumes have been obtained, we can start the backtracking process to obtain the 3D shortest path. One 3D path is extracted for each horizontal scanline of the input image.

### C. Sub-pixel Accuracy

The result of shortest path extraction produces motion estimation up to pixel level accuracy. Sub-pixel accuracy can be obtained by fitting a second degree surface to the correlation coefficients in the

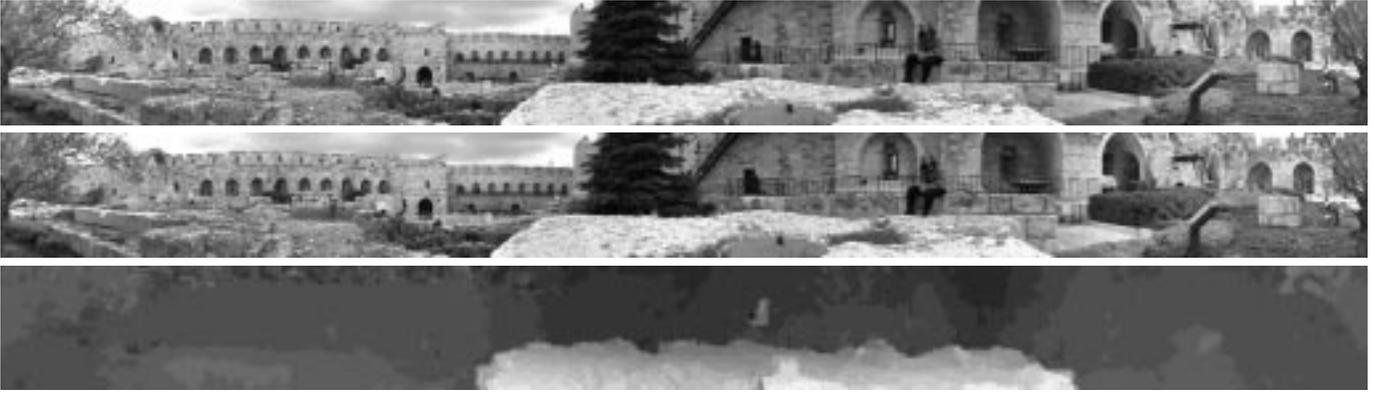


Fig. 10. The first and second images are the left and right input images [20]. The third image gives the matching results using our method described in this paper.

neighbourhood of the motion vector and the extrema of the surface can be obtained analytically. The general form of the second degree surface is:  $S(x, y) = A \cdot x^2 + B \cdot xy + C \cdot y^2 + D \cdot x + E \cdot y + F$ . The maximum can be found where the slope is zero in the quadratic equation. The sub-pixel position can be found by solving the following equation:

$$\begin{cases} 2A \cdot x + B \cdot y + D = 0 \\ B \cdot x + 2C \cdot y + E = 0 \end{cases} \quad (6)$$

Solving Eq. (6) we have:

$$\begin{cases} x = (BE - 2CD)/(4AC - B^2) \\ y = (BD - 2AE)/(4AC - B^2) \end{cases} \quad (7)$$

When estimating the coefficients  $A, B, C, D, E$  and  $F$  of function  $S(x, y)$ , one usually needs to solve a set of over-determined linear equations. A quick way of obtaining the coefficients of  $S(x, y)$  is necessary to make sub-pixel accuracy motion estimation practical.

If the shortest path passes position  $(p, q)$  at plane  $k$  of the volume, we use the nine correlation coefficient values in the neighbourhood of  $(p, q)$  as input. We have derived the following formula for the calculation of  $A, B, C, D, E$  and  $F$  using nine neighbouring values.

$$\begin{cases} A = (b_{(-,-)} - 2b_{(-,+)} + b_{(+,-)} + b_{(-,.)} - 2b_{(.,.)} + b_{(+,.)} + \\ \quad b_{(-,+)} - 2b_{(.,+)} + b_{(+,+)})/6 \\ B = (b_{(-,-)} - b_{(+,-)} - b_{(-,+)} + b_{(+,+)})/4 \\ C = (b_{(-,-)} + b_{(-,+)} + b_{(+,-)} - 2b_{(-,.)} - 2b_{(.,.)} - 2b_{(+,.)} + \\ \quad b_{(-,+)} + b_{(.,+)} + b_{(+,+)})/6 \\ D = (-b_{(-,-)} + b_{(+,-)} - b_{(-,.)} + b_{(+,.)} - b_{(-,+)} + b_{(+,+)})/6 \\ E = (-b_{(-,-)} - b_{(-,+)} - b_{(+,-)} + b_{(-,+)} + b_{(+,+)})/6 \\ F = (-b_{(-,-)} + 2b_{(-,+)} - b_{(+,-)} + 2b_{(-,.)} + 5b_{(.,.)} + 2b_{(+,.)} - \\ \quad b_{(-,+)} + 2b_{(.,+)} - b_{(+,+)})/9 \end{cases} \quad (8)$$

where  $b_{(-,-)} = S(p-1, q-1)$ ,  $b_{(-,.)} = S(p, q-1)$ ,  $b_{(+,-)} = S(p+1, q-1)$ ,  $b_{(-,+)} = S(p-1, q)$ ,  $b_{(.,.)} = S(p, q)$ ,  $b_{(+,.)} = S(p+1, q)$ ,  $b_{(-,+)} = S(p-1, q+1)$ ,  $b_{(.,+)} = S(p, q+1)$ ,  $b_{(+,+)} = S(p+1, q+1)$ . The  $b_{(*,*)}$ 's are the values of the local correlation coefficients. One can, therefore, use Eq. (8) to obtain the coefficients of function  $S(x, y)$ , and then use Eq. (7) to calculate the sub-pixel accuracy motion vector.

#### D. Algorithm Steps

The steps of our algorithm for fast image motion estimation are:

- 1) Performing fast ZNCC (or use SSD or SAD) to obtain the correlation coefficients;
- 2) Building a 3D correlation coefficient volume for each scanline (or each column) of the image;

- 3) Using dynamic programming technique to find the best path in the 3D volume, which will then give the motion vectors;
- 4) Fitting the correlation values in the neighbourhood of the motion vector obtained in the previous step to a surface to obtain sub-pixel accuracy.

#### E. Experimental Results

This section shows some of the results obtained using our motion estimation method. Comparisons with some of the commonly cited techniques are also made. A variety of images have been tested, including synthetic images and different types of real images.

Fig. 13 shows the results of different techniques on the image sequence "Yosemite". The first two images in the top row are frames 9 and 10 in the sequence. The third picture in the top row is the correct optical flow field. The results of Fleet's, Horn's and Lucas' techniques give sparse flow fields, while other techniques give dense optical flow. The techniques producing reasonable results for the top region of the image are Singh's and ours.

Table IV shows the errors, flow density, number of image frames used and the time that several techniques used for calculating the flow field. The errors in Fleet's, Horn's and Lucas' techniques are small because they only use the reliable flow estimates. Uras' technique and our technique give smaller errors and our technique gives the higher computation speed. But Uras *et al*'s technique does not perform well at the top region of the image, and 15 frames of the sequence are required. The test were run on a 85MHz Sun SPARCserver1000 running Solaris 2.5. All the programs apart from the author's were obtained from the ftp site at <ftp://csd.uwo.ca/pub/vision>. The typical running time for our algorithm on a  $256 \times 256$  image is in the order of seconds.

TABLE IV  
RESULTS FOR THE IMAGE SEQUENCE yos.

Technique	Av. error	Standard deviation	Density	Frms used	User time
Anandan	16.37	13.46	100.00%	2	849.79s
Fleet	5.28	14.33	30.64%	15	426.13s
Horn	5.48	11.30	32.88%	15	29.62s
Lucas	4.48	12.16	39.78%	15	32.94s
Nagel	12.70	16.68	100.00%	15	205.50s
Quenot	9.93	16.16	100.00%	2	182.63s
Singh	12.09	15.86	100.00%	3	339.36s
Uras	8.92	15.61	100.00%	15	17.58s
Sun	9.21	16.16	100.00%	2	14.35s

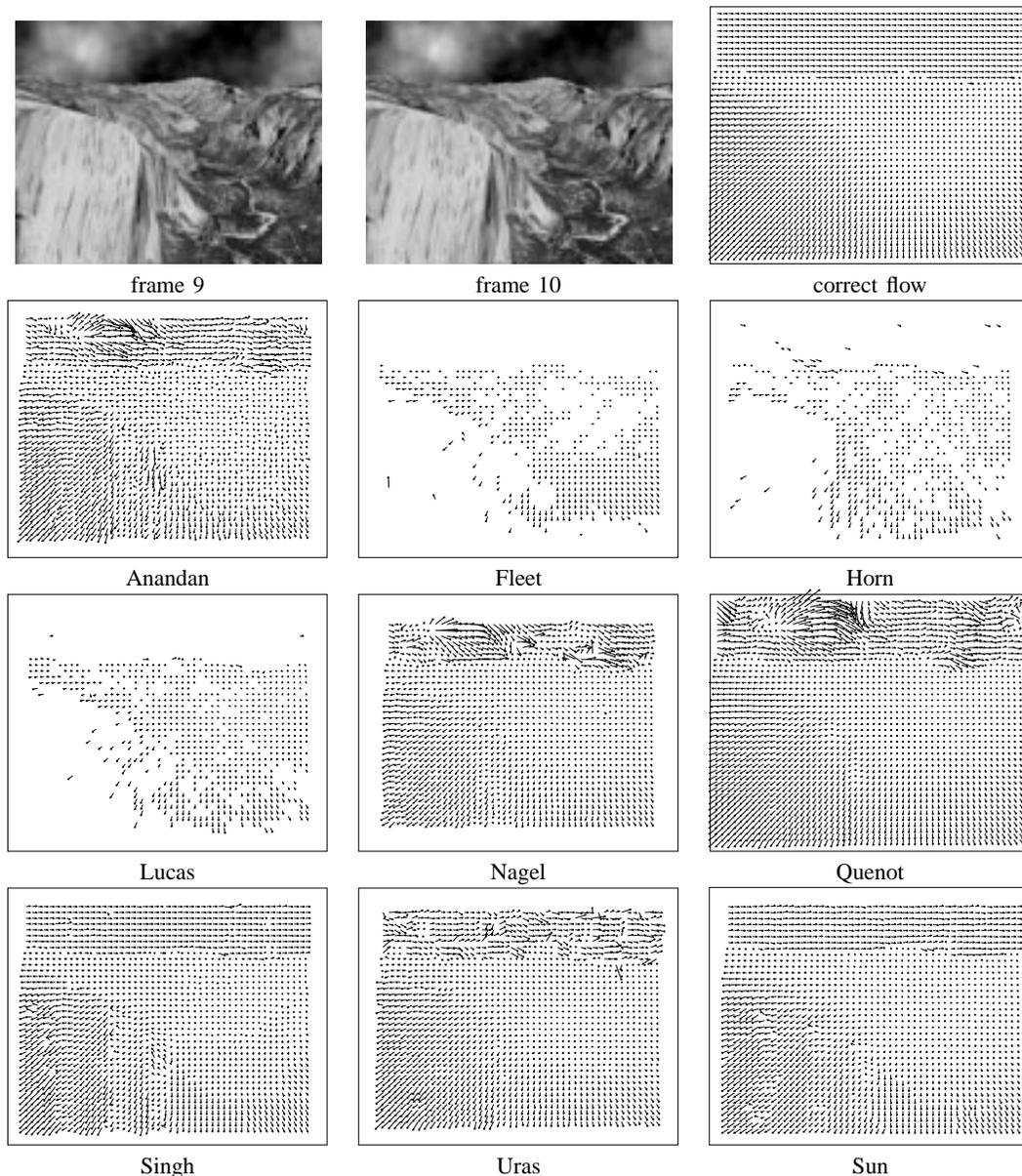


Fig. 13. The optical flow results of different techniques on the **Yosemite** sequence. The first two images in the first row are the frames 9 and 10 in the sequence, and the third picture in the first row shows the true optical flow. The name of each technique is given below the corresponding picture.

Four real image sequences have also been tested, and good results have been obtained. Fig. 14 shows the results of several techniques on the four real image sequences: SRI Trees, NASA Sequence, Rubik Cube and Hamburg Taxi provided in [9].

#### F. Fast Image Motion Web Demo

The web demo address is at:  
<http://extra.cmis.csiro.au/IA/changs/motion/>

#### VI. DISCUSSION ON RELIABILITY AND COMPUTATIONAL SPEED

The reliable results of our algorithm are achieved by applying the combination of the following techniques: (1) Coarse-to-fine strategy is used (for stereo matching). (2) The zero mean normalized cross correlation similarity measure is used, which is independent of differences in brightness and contrast. (3) The correlation coefficient value is used as input to the dynamic programming stage rather than just using the intensity value of the input images. (4) Dynamic

programming technique is used to find a 3D maximum-surface or 3D shortest path in the correlation volume.

The fast computational speed of our algorithm is achieved with the following aspects: (1) Fast zero mean normalized cross correlation is used. (2) We have used a rectangular subregioning technique for fast computation of correlation coefficients (for stereo matching). (3) Apart from having the advantages of increasing the reliability, the coarse-to-fine approach is also faster than one without using it. (4) A two-stage dynamic programming technique is used to find a maximum-surface in the 3D correlation volume. (5) A simple formula is used for sub-pixel motion estimation.

#### VII. CONCLUSIONS

We have developed a fast and reliable stereo matching method using rectangular subregioning, fast correlation and 3D maximum-surface techniques in the coarse-to-fine framework. The 3D maximum-surface is obtained from the 3D correlation volume using a two-stage dynamic programming technique. A similar technique

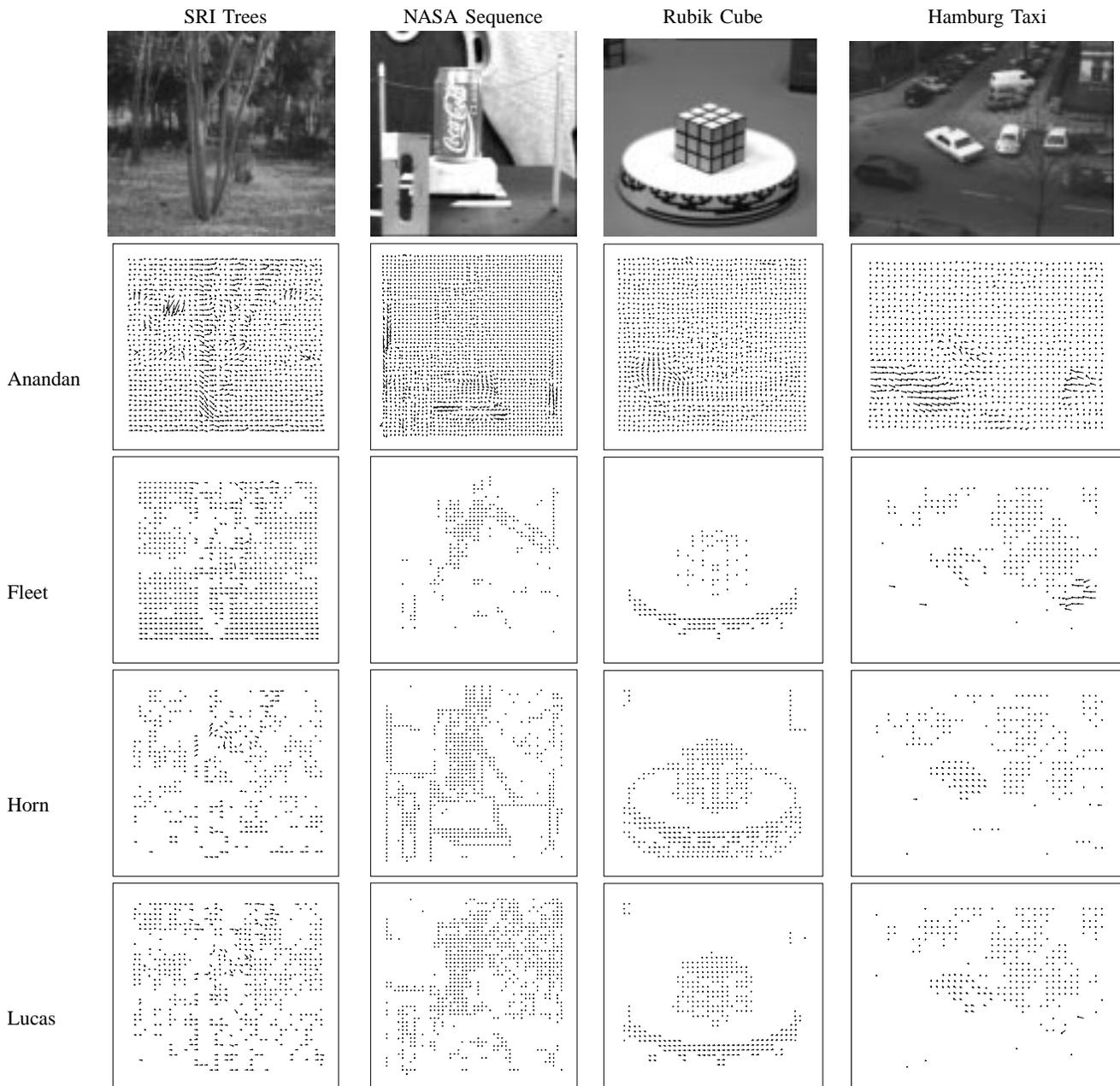


Fig. 14. The results of different techniques on four of the commonly used images sequences. (Images courtesy of Barron, Fleet and Beauchemin [9].)

was developed for fast panoramic stereo matching using cylindrical maximum surface techniques. We also developed an image motion estimation method using fast correlation and 3D shortest path techniques. All the algorithms were shown to be fast and reliable by testing on several different types of real images.

#### ACKNOWLEDGEMENT

The panoramic stereo images used in this paper were acquired at the School of Computer Science and Engineering, The Hebrew University, 91904 Jerusalem, Israel. The author is grateful to the owners of the images used in this paper.

#### REFERENCES

- [1] C. Sun, "Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques," *International Journal of Computer Vision*, vol. 47, no. 1/2/3, pp. 99–117, April-June 2002.
- [2] C. Sun and S. Peleg, "Fast panoramic stereo matching using cylindrical maximum surfaces," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 2003, to appear.
- [3] C. Sun, "Fast optical flow using 3D shortest path techniques," *Image and Vision Computing*, vol. 20, no. 13/14, pp. 981–991, December 2002.
- [4] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," *International Journal of Computer Vision*, vol. 35, no. 3, pp. 269–293, 1999.
- [5] D. Scharstein and R. Szeliski, "Stereo matching with nonlinear diffusion," *International Journal of Computer Vision*, vol. 28, no. 2, pp. 155–174, 1998.
- [6] Y.-S. Kim, J.-J. Lee, and Y.-H. Ha, "Stereo matching algorithm based on modified wavelet decomposition process," *Pattern Recognition*, vol. 30, no. 6, pp. 929–952, 1997.
- [7] B. Porr, A. Cozzi, and F. Wörgötter, "How to 'hear' visual disparities: real-time stereoscopic spatial depth analysis using temporal resonance," *Biological Cybernetics*, vol. 78, no. 5, pp. 329–336, 1998.
- [8] D. G. Jones and J. Malik, "Computational framework for determining stereo correspondence from a set of linear spatial filters," *Image and*

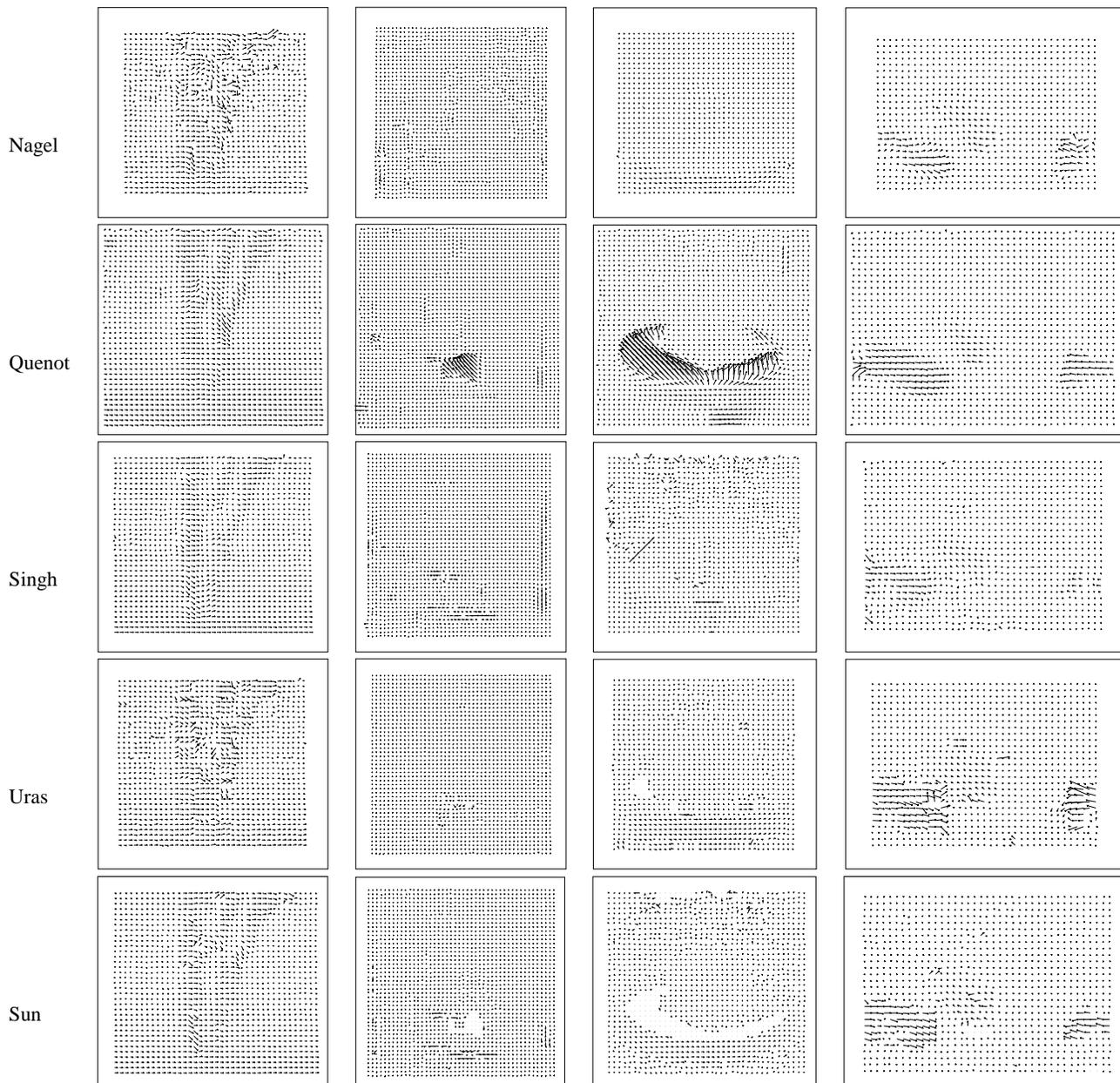


Fig. 14. (cont'd) The results of different techniques on four of the commonly used images sequences. (Images courtesy of Barron, Fleet and Beauchemin [9].)

*Vision Computing*, vol. 10, no. 10, pp. 699–708, December 1992.

- [9] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.
- [10] M. Rechsteiner, B. Schneuwly, and G. Troester, "Dynamic workspace monitoring," in *International Archives of Photogrammetry and Remote Sensing*, H. Ebner, C. Heipke, and K. Eder, Eds., vol. 30, Munich, Germany, September 1994, pp. 689–696.
- [11] P. Aschwandner and W. Guggenbühl, "Experimental results from a comparative study on correlation-type registration algorithms," in *Robust Computer Vision*, W. Förstner and S. Ruwiedel, Eds. Wichmann, 1992, pp. 268–289.
- [12] O. Faugeras, B. Hotz, H. Mathieu, T. Viéville, Z. Zhang, P. Fua, E. Théron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy, "Real time correlation-based stereo: Algorithm, implementations and applications," INRIA, Tech. Rep. RR-2013, 1993.
- [13] C. Sun, "A fast stereo matching method," in *Digital Image Computing: Techniques and Applications*, Massey University, Auckland, New Zealand, 10–12 December 1997, pp. 95–100.
- [14] —, "Multi-resolution rectangular subregioning stereo matching using fast correlation and dynamic programming techniques," CSIRO Mathematical and Information Sciences, Australia, Tech. Rep. 98/246, December 1998.
- [15] S. Roy, "Stereo without epipolar lines: A maximum-flow formulation," *International Journal of Computer Vision*, vol. 34, no. 2/3, pp. 147–161, 1999.
- [16] I. Cox, S. Hingorani, S. Rao, and B. Maggs, "A maximum likelihood stereo algorithm," *Computer Vision and Image Understanding*, vol. 63, no. 3, pp. 542–567, May 1996.
- [17] W. Hoff and N. Ahuja, "Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 2, pp. 121–136, February 1989.
- [18] S. Peleg and M. Ben-Ezra, "Stereo panorama with a single camera," in *Proceedings of Computer Vision and Pattern Recognition*, Ft. Collins, Colorado, June 1999, pp. 395–401.
- [19] H.-Y. Shum and R. Szeliski, "Stereo reconstruction from multiperspective panoramas," in *Proceedings of International Conference on Computer Vision*, vol. I, Kerkyra, Greece, 20–27 September 1999, pp. 14–21.
- [20] S. Peleg, M. Ben-Ezra, and Y. Pritch, "Omnistere: Panoramic stereo imaging," *IEEE Transactions on Pattern Analysis and Machine Intelli-*

- gence, vol. 23, no. 3, pp. 279–290, March 2001.
- [21] J. Gluckman, S. K. Nayar, and K. J. Thoresz, “Real-time omnidirectional and panoramic stereo,” in *DARPA Image Understanding Workshop*. Monterey, California: Morgan Kaufmann, November 1998, pp. 299–303.
  - [22] H. Bakstein and T. Pajdla, “Omnivergent stereo-panoramas with a fish-eye lens,” Center for Machine Perception, Czech Technical University, Tech. Rep. CTU-CMP-2001-22, August 2001.
  - [23] M. Fiala and A. Basu, “Panoramic stereo reconstruction using non-SVP optics,” in *Proceedings of International Conference on Pattern Recognition*, Quebec City, Canada, 11-15 August 2002.
  - [24] H.-Y. Shum, A. Kalai, and S. M. Seitz, “Omnivergent stereo,” in *Proceedings of International Conference on Computer Vision*, vol. I, Kerkyra, Greece, 20-27 September 1999, pp. 22–29.
  - [25] J. Y. Zheng and S. Tsuji, “Panoramic representation for route recognition by a mobile robot,” *International Journal of Computer Vision*, vol. 9, no. 1, pp. 55–76, 1992.
  - [26] Y. Li, C. K. Tang, and H. Y. Shum, “Efficient dense depth estimation from dense multiperspective panoramas,” in *Proceedings of International Conference on Computer Vision*, Vancouver, Canada, 9-12 July 2001, pp. 119–126.
  - [27] C. Sun and S. Pallottino, “Circular shortest path in images,” *Pattern Recognition*, vol. 36, no. 3, pp. 711–721, March 2003.
  - [28] G. L. Gimel’farb, V. M. Krot, and M. V. Grigorenko, “Experiments with symmetrized intensity-based dynamic programming algorithms for reconstructing digital terrain model,” *International Journal of Imaging Systems and Technology*, vol. 4, no. 1, pp. 7–21, 1992.
  - [29] A. F. Bobick and S. S. Intille, “Large occlusion stereo,” *International Journal of Computer Vision*, vol. 33, no. 3, pp. 181–200, 1999.
  - [30] B. Appleton and C. Sun, “Circular shortest paths by branch and bound,” *Pattern Recognition*, 2003, to appear.
  - [31] P. Anandan, “A computational framework and an algorithm for the measurement of visual motion,” Computer and Information Science, University of Massachusetts at Amherst, Tech. Rep. 87-73, August 1987.
  - [32] —, “A computational framework and an algorithm for the measurement of visual motion,” *International Journal of Computer Vision*, vol. 2, pp. 283–310, 1989.
  - [33] A. Singh, “An estimation-theoretic framework for image flow computation,” in *Proceedings of International Conference on Computer Vision*, Osaka, 1990, pp. 168–177.
  - [34] —, *Optic Flow Computation: A Unified Perspective*. IEEE Computer Society Press, 1992.
  - [35] D. J. Heeger, “Optical flow using spatiotemporal filters,” *International Journal of Computer Vision*, vol. 1, pp. 279–302, 1988.
  - [36] D. J. Fleet and A. D. Jepson, “Computation of component image velocity from local phase information,” *International Journal of Computer Vision*, vol. 5, pp. 77–104, 1990.
  - [37] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, pp. 185–204, 1981.
  - [38] B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Image Understanding Workshop*, 1981, pp. 121–130.
  - [39] S. Uras, F. Girosi, A. Verri, and V. Torre, “A computational approach to motion perception,” *Biological Cybernetics*, vol. 60, pp. 79–97, 1988.
  - [40] H.-H. Nagel, “On the estimation of optical flow: Relations between different approaches and some new results,” *Artificial Intelligence*, vol. 33, pp. 299–324, 1987.
  - [41] G. M. Qu  not, “The ‘orthogonal algorithm’ for optical flow detection using dynamic programming,” in *International Conference on Acoustics, Speech and Signal Processing*, vol. 3, San Francisco, CA, March 1992, pp. 249–252.