Digital Image Computing: Techniques and Applications.
Perth, Australia, December 7-8, 1999, pp.195-200.

Multi-Resolution Stereo Matching Using Maximum-Surface Techniques

Changming Sun

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

Abstract

This paper presents a fast and reliable stereo matching algorithm which produces a dense disparity map by using fast cross-correlation, rectangular subregioning and maximum-surface techniques in a coarse-to-fine scheme. Fast correlation is achieved by using the box filtering technique whose speed is invariant to the size of the correlation window, and by segmenting the images at different levels of the pyramid into rectangular subimages. By working with rectangular subimages, the speed of the algorithm can be increased and the intermediate memory storage requirement is reduced. The disparity for the whole image is found in the correlation coefficient volume by obtaining the maximum-surface rather than simply choosing the position that gives the maximum correlation coefficient value. Typical running time for a 512×512 image is in the order of half a minute rather than minutes or hours. A variety of synthetic and real images have been tested, and good results have been obtained.

1. Introduction

The correspondence problem in stereo vision and photogrammetry concerns the matching of points or other kinds of primitives such as edges and regions in two images such that the matched points are the projections of the same point in the scene. The disparity map obtained from the matching stage may then be used to compute the 3D positions of the scene points given knowledge about the relative orientation of the two cameras.

Intille and Bobick [8] presented a stereo algorithm that incorporates the detection of the occlusion regions directly into the matching process. They developed a dynamic programming solution that obeys the occlusion and ordering constraints to find a best path through the disparity-space image. Fua [6] described a correlation based multi-resolution algorithm which was followed by interpolation. Anandan [2] described a hierarchical computational framework for the determination of dense motion fields using correlation-based method from a pair of images. A number of researchers have also used dynamic programming to solve globally the matching problem [7, 10].

Sun [13, 14] developed a fast stereo matching method using fast cross correlation and dynamic programming techniques. The dynamic programming was applied to the correlation coefficients matrix along the corresponding epipolar lines. He did not consider the continuity of neighbouring epipolar lines. Roy and Cox [12] developed an algorithm for solving the N-camera stereo correspondence problem by transforming it into a maximum-flow problem. The minimum-cut associated to the maximum-flow yielded a disparity surface for the whole image at once. The *preflow-push lift-to-front* algorithm was used when they calculated the maximum-flow. The average running time for Roy and Cox's algorithm was $O((MN)^{1.2}N^{1.3})$ (with image size M, N and depth resolution D) [12]. Chen and Medioni [4] presented a propagation type of algorithm similar to [10]. The techniques they used included non-maxima suppression, seed voxel selection and surface tracing. There was no mention of the speed issues in [4].

In this paper we address some of the efficient and reliable aspects of the stereo matching algorithms by using fast correlation, rectangular subregioning and maximum-surface techniques in a multi-resolution scheme. The disparity is obtained from the 3D correlation coefficient volume using a two-stage dynamic programming technique considering the continuity of the neighbouring epipolar scan lines. The combination of these techniques results in very fast and reliable stereo matching. The rest of the paper is organised as follows: Section 2 describes the method for fast calculation of similarity measure. Section 3 presents our new method of stereo matching by finding the maximum surface in the 3D correlation volume by using dynamic programming techniques. The detailed matching strategy is described in Section 4. Section 5 shows the experimental results obtained using our fast stereo matching method applied to a variety of images. Section 6 discusses the reliability and computation speed issues of our algorithm. Section 7 gives concluding remarks.

2. Fast Calculation of Similarity Measure

Similarity is the guiding principle for solving the correspondence problem. Corresponding features or areas should be similar in the two images. Different similarity measures have been used in the literature for matching, and their performance and computation cost vary [11, 3]. The most commonly used similarity measure is the cross-correlation coefficient. We will use the zero-mean normalised cross-correlation (ZNCC) coefficient as the measure of similarity between the source and the candidate matching areas. The estimate is independent of differences in brightness and contrast due to the normalisation with respect to mean and standard deviation. But direct calculation of ZNCC is computationally expensive compared with the sum of absolute difference or the sum of squared difference. Faugeras *et al* [5] 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 our early work in [13, 14] for achieving fast correlation.

2.1. Fast Cross-Correlation

Let $f_{m,n}$ be the intensity value of an $M \times N$ image f at position (m,n), where f is to be locally averaged into $\bar{f}$, i.e. obtaining the mean of the original image within a box. We also have similar definition for a second image g. The zero-mean normalised cross-correlation of two local windows can be written as follows:

$$C(i,j,d) = \frac{cov_{ij,d}(f,g)}{var_{ij}(f) \times var_{ij,d}(g)} \quad (1)$$

where

$$cov_{ij,d}(f,g) = \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} (f_{m,n} - \bar{f}_{i,j})(g_{m+d,n} - \bar{g}_{i+d,j}) \quad (2)$$

and i, j are the image row and column indices. d is the shift of the window along epipolar lines, which indicates the possible disparity values; K and L define the correlation window size. $\bar{f}$ and $\bar{g}$ are the mean values within the windows. $var_{ij}(f)$ and $var_{ij,d}(g)$ are the square roots of the variances for the left and right local windows. Using the algorithms described in [13, 14], the cross correlation coefficients can be obtained efficiently.

The result of the correlation calculation described above is a 3D volume containing the correlation coefficients as shown in Fig. 1. The size of the volume depends upon the image row and column numbers M, N and the maximum disparity search range D. The complexity of the algorithm is $O(MND)$. The storage space needed for the correlation coefficients is in the order of $4MND$ bytes.

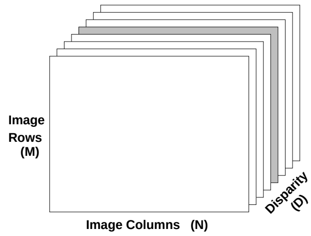

Figure 1. An illustration of the 3D correlation coefficient volume obtained after using the fast correlation method.

2.2. Using Subimages

Rather than working with the whole image during the fast image correlation stage as described in the previous subsections, we could work with subimages to speed up the correlation calculation further and reduce the memory space for storing the correlation coefficients. As mentioned earlier, the computation complexity for the fast image correlation step is MND if we work with the whole image.

If the image is divided into n subimages or rectangular subregions, the computation complexity will be $\sum_{i=0}^{n-1}(M_i N_i D_i)$, where M_i, N_i are the row and column numbers for the ith subimage or region, and D_i is the disparity search range over this subimage. It is anticipated that $\sum_{i=0}^{n-1}(M_i N_i D_i)$ will be smaller than MND, especially when the disparity changes a lot within the whole image. When actually performing correlation calculation for each of the subregions, certain size of region overlapping needs to be considered in order to eliminate the boundary effect. It is also necessary to allow some overlapping between successive horizontal stripes. The amount of overlapping depends on the size of the correlation window used. When calculating the corresponding positions of a subregion in the right image after knowing the position in the left image, the disparity information of this region in the disparity map will be used. For detailed description see our early work [14].

There is another advantage for working with subimages in terms of memory usage. As mentioned in the previous subsections, some memory space is needed to store the correlation coefficients. In the case of working with one whole image, the memory space needed is in the order of $4MND$ bytes. While in the case of working with subimages, the memory space needed is in the order of $\max_i(4M_i N_i D_i)$, because the memory for each subregion is dynamically allocated and freed.

3. Maximum-Surface in the Volume

Sun [13] and Intille and Bobick [8] chose a slice of the correlation coefficient volume as a 2D correlation matrix for each scan line of the input image and use this matrix to obtain disparities. As mentioned earlier in the Introduction section, Roy and Cox [12] and Chen and Medioni [4] used 3D volume information to find disparities.

In this section, we will approach the issue of obtaining disparity map from 3D correlation coefficient volume using dynamic programming techniques, which is computationally efficient. A maximum surface which cuts through the 3D volume from the top to the bottom as shown in Fig. 2 is obtained using a two-stage dynamic programming technique. The maximum-surface gives the maximum sum of the correlation coefficients along the surface when certain constraints are imposed. The disparity gradient limit constraint can be easily implemented during the dynamic programming process. This limit constrains the size of neighbourhood search or the surface along which it can go.

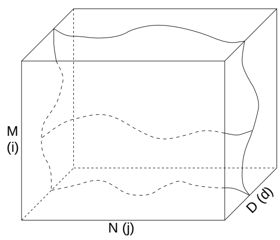

Figure 2. The maximum surface which give the maximum accumulation of values in the cross correlation coefficient volume.

Now we describe our novel algorithm for the maximum-surface extraction in a 3D volume of size MND. Assume $C(i,j,d)$ is the correlation coefficient value in the 3D volume at position (i,j,d), where $0 \leqslant i < M, 0 \leqslant j < N$, and $0 \leqslant d < D$. Array $Y(i,j,d)$ contains the accumulated values of the maximum cross correlation coefficients along all the possible surfaces in the same volume from top to bottom. For the top horizontal slice of the volume, i.e. when $i = 0$,

$$Y(0,j,d) = C(0,j,d) \tag{3}$$

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 using the following recursion:

$$Y(i,j,d) = C(i,j,d) + \max_{t:|t| \leqslant p} Y(i-1,j,d+t) \tag{4}$$

where p determines the number of local values that need to be checked. If $p = 1$, only three values in Y need to be evaluated. The three values are $Y(i-1,j,d-1), Y(i-1,j,d)$ and $Y(i-1,j,d+1)$.

After the recursion described in the previous paragraph, $Y(i,j,d)$ contains the maximum sum of $C(i,j,d)$ from top to bottom of the 3D volume. We now use volume Y to obtain the disparity map for the input stereo images. Starting from the bottom of the 3D volume Y, we select the 2D horizontal slice with $i = M$-1. From this 2D matrix of size ND, a shortest-path from left to right is obtained using dynamic programming techniques, as illustrated by the dotted line inside the shaded region in Fig. 3. The summation of the values along this path gives the maximum value. 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 dashed line in Fig. 3 is the obtained disparity for the same x- positioned point of the input image.

When calculating the disparity for row number i-1, we use the result obtained for row number i. We now select the horizontal slice number i-1 of the 3D volume Y, and mask out those values outside the grey region which are p position away from the shortest-path obtained from row number i, as shown in Fig. 3. Then a new shortest-path (the black curve) is obtained in this 2D matrix from left to right which are constrained to lie inside this grey region. This process of obtaining shortest-path is repeated until the disparity for the first row of the image is obtained.

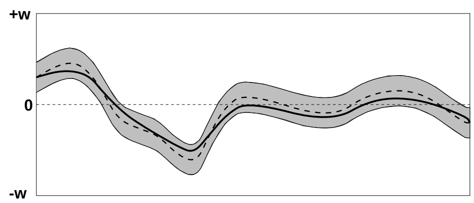

Figure 3. An illustration of the shortest-path obtained for each horizontal slice of the $Y(i,j,d)$ volume.

Putting the shortest-paths for each of the scan line together form 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 surface gives more consistent disparities.

4. Matching Strategy

4.1. Coarse-to-fine Scheme

It has been shown that a multi-resolution or pyramid data structure approach to stereo matching is faster than one without multi-resolution [9], as the search range in each level is small. Besides fast computation, a more reliable disparity map can be obtained by exploiting the multi-resolution data structure. The upper levels of the pyramids are ideal to get an overview of the image scene. The details can be found down the pyramid at higher resolution. There are three useful properties for the coarse-to-fine scheme [1]: (a) the pull-in range or search range can be increased, because at a coarse pyramidal level only rough initial values are needed; (b) the convergence speed can be improved; and (c) the reliability of finding correct matches can be increased.

4.2. Sub-pixel Accuracy

Sub-pixel accuracy can be obtained by fitting a second degree curve to the correlation coefficients in the neighbourhood of the disparity and the extrema of the curve can be obtained analytically. The general form of the second degree curve (parabola) is: $f(x) = a + b \cdot x + c \cdot x^2$. The maximum can be found where the slope is zero in the quadratic equation. The sub-pixel position can be found at $x = -b/2c$. If only three points of the correlation values are used, e.g. the points at $i-1, i, i+1$, the sub-pixel position of the disparity can be calculated using the following formula [2]:

$$x = i + \frac{1}{2} \times \frac{C(i-1) - C(i+1)}{C(i-1) - 2C(i) + C(i+1)} \quad (5)$$

where $C(i)$ is the correlation value in the matrix at position i, and x is the sub-pixel disparity obtained. If five points of the correlation values are used, e.g. the five points at $i-2, i-1, i, i+1, i+2$, we derive the following equation for the calculation of sub-pixel position:

$$x = i + \frac{7}{20} \times \frac{2C(i-2) + C(i-1) - C(i+1) - 2C(i+2)}{2C(i-2) - C(i-1) - 2C(i) - C(i+1) + 2C(i+2)} \quad (6)$$

4.3. Algorithm Steps

Our proposed algorithm for stereo matching is:

1. Build image pyramids with K levels (from 0 to $K-1$);
2. Initialise the disparity map as zero for level $k = K-1$ and start stereo matching at this level;
3. Perform image matching using the method described in Sections 2-4 which includes:
   (a) Segment images into rectangular subregions;
   (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 maximum surface, which will then give the disparity map as described in Section 3.
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; otherwise go to Step 5;
5. Fit curve to obtain sub-pixel accuracy using Eq. (5) or Eq. (6) if necessary.
6. Display disparity map.

5. Experiment Results

This section shows some of the results obtained using our new method described in this paper. A variety of images have been tested, including synthetic images and different types of real images. The input left and right images are assumed to be rectified epipolar images.

Synthetic Images

Fig. 4 gives the result of the algorithm running on two pairs of synthetic images. The two columns on the left show the input left and right images. The third column is the results obtained using our earlier method presented in [13, 14]. The last column shows the results using the method described in this paper. The top row of the figure shows a concrete sphere on a table. The sizes of both of these images are 256×256. The left hand side of Fig. 4(c,d) contains a stripe of black region which indicates that this region in the left image does not have corresponding pixels in the right image. The bottom row shows images of a corridor and the results. The size of the corridor images is 512×512. It can be seen that our new method gives better results.

Real Images

Many other types of real images have been tested, and good results have been obtained. Due to limitation of space, only small portion of the tested images were shown here. Fig. 5 gives some of the results obtained by using the methods in [13, 14] and our new method described in this paper. Comparing the results in the third and the last columns of the figure, it can be seen that our new method gives more reliable matching results.

Running Times

Table 1 gives some of the typical running times of the algorithm on different size of images with different disparities. The tests were run on an 85MHz Sun SPARCserver1000 running Solaris 2.5. The typical running time for the algorithm on a 256×256 image is in the order of seconds rather than minutes or even hours. The size of the correlation window used for the images given in the table is 9×9. The reduction ratio r used in the pyramid generation process is 2. The last two columns in the table show the timing of the algorithm described in [14] (Method 2D path) and

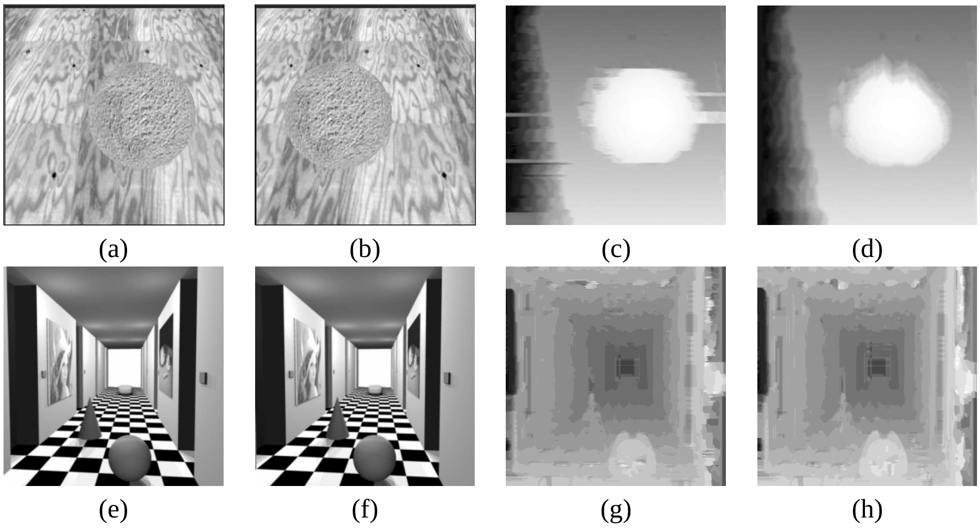

Figure 4. The matching result for synthetic images. Top row gives the images of a sphere on a table. The bottom row shows the images of a corridor. (a,e) left image; (b,f) right image; (c,g) the disparity map recovered using method in [13, 14]; and (d,h) the disparity map recovered using our new method.

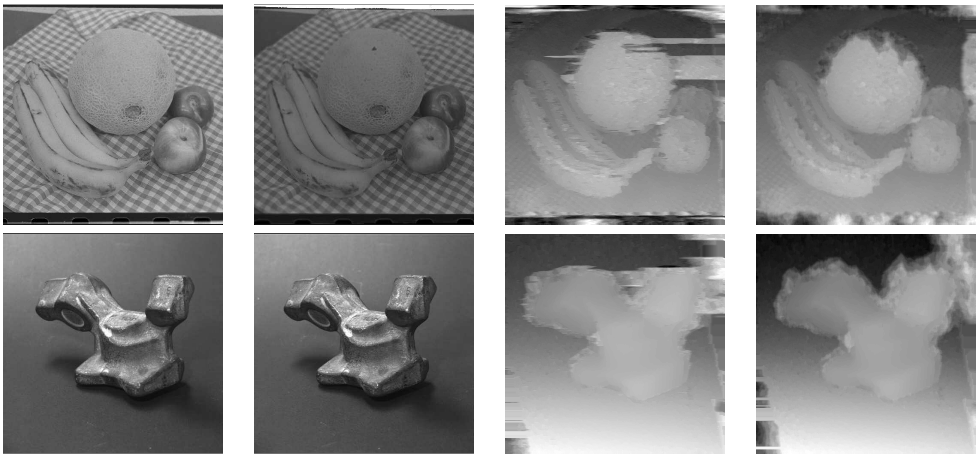

Figure 5. The matching result for some real images. The first and second columns are the left and right input images. The third column gives the matching results using the method described in [13, 14]. The last column is the results obtained using our new method.

the algorithm described in this paper (Method 3D). There is not much difference in the speed of the two algorithms. But the 3D maximum-surface method developed in this paper gives more reliable results. Interested readers could try their own images by accessing the following web page: `http://www.dms.csiro.au/~changs/cgi-bin/`

6. 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. (2) The zero-mean normalised cross-correlation similarity measure is used. (3) The correlation coefficient value is used as input to the dynamic programming stage. (4) Dynamic programming technique is used to find a maximum-surface in the correlation

Table 1. Running times of the algorithm on different images. (Image names: A=ball, B=pentagon, C=flat. Third column is the pyramid levels used.)

	Image size	Pyr. lvls	Disparity range	Method 2D path	Method 3D
A	256×256	2	[-19,7]	4.70s	5.29s
B	512×512	3	[-13,12]	25.05s	21.82s
C	1000×1000	4	[-39,27]	140.74s	136.25s

volume. By using the dynamic programming technique on the input correlation coefficient volume, one will obtain a more smooth surface within the volume. The maximum surface method takes all the information into account, rather than work individually for each of the epipolar lines.

The fast computational speed of our algorithm is achieved in conjunction with some of the aspects mentioned above for achieving reliability of the algorithm. Some of the aspects are: (1) Fast zero-mean normalised cross correlation is developed. (2) We have used a rectangular subregioning technique for fast computation of correlation coefficients. (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. Rather than using the methods described in [12, 4], a dynamic programming technique is used which is computationally efficient.

7. Conclusions

We have developed a fast and reliable stereo matching method using rectangular subregioning, fast correlation and maximum-surface techniques in the coarse-to-fine framework. The maximum-surface is obtained from the 3D correlation volume using a two-stage dynamic programming technique. The algorithm produces a reliable dense disparity map. The fast cross-correlation method was developed from the box-filtering idea. The time spent in the stage for obtaining the normalised cross-correlation is almost invariant to the search window size. The processing speed is further improved by segmenting the input image into subimages and work with the smaller images which tend to have smaller disparity ranges. The subregioning technique is also helpful to reduce the memory storage space. The typical running time for a 512×512 image is in the order of half a minute rather than minutes or hours. The algorithm is shown to be fast and reliable by testing on several different types of images: both synthetic and real images.

References

[1] F. Ackermann and M. Hahn. Image pyramids for digital photogrammetry. In H. Ebner, D. Fritsch, and C. Heipke, editors, *Digital Photogrammetric Systems*, pages 43–58. Wichmann, 1991.

[2] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[3] P. Aschwanden and W. Guggenbühl. Experimental results from a comparative study on correlation-type registration algorithms. In W. Förstner and S. Ruwiedel, editors, *Robust Computer Vision*, pages 268–289. Wichmann, 1992.

[4] Q. Chen and G. Medioni. Building human face models from two images. In *Multimedia Signal Processing*, Redonda Beach, CA, 1998.

[5] 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. Technical Report RR-2013, INRIA, 1993.

[6] P. Fua. A parallel stereo algorithm that produce dense depth maps and preserves image features. *Machine Vision Applications*, 6:35–49, 1993.

[7] 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*, 4:7–21, 1992.

[8] S. S. Intille and A. F. Bobick. Disparity-space images and large occlusion stereo. In *Proceedings of European Conference on Computer Vision*, Stockholm, Sweden, 1994.

[9] K. S. Kumar and U. B. Desai. New algorithms for 3D surface description from binocular stereo using integration. *Journal of the Franklin Institute*, 331B(5):531–554, 1994.

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

[11] M. Rechsteiner, B. Schneuwly, and G. Troester. Dynamic workspace monitering. In H. Ebner, C. Heipke, and K. Eder, editors, *International Archives of Photogrammetry and Remote Sensing*, volume 30, pages 689–696, Munich, Germany, September 1994.

[12] S. Roy and I. J. Cox. A maximum-flow formulation of the N-camera stereo correspondence problem. In *Proceedings of International Conference on Computer Vision*, pages 492–499, Bombay, India, January 1998. IEEE.

[13] C. Sun. A fast stereo matching method. In *Digital Image Computing: Techniques and Applications*, pages 95–100, Massey University, Auckland, New Zealand, December 10-12 1997.

[14] C. Sun. Multi-resolution rectangular subregioning stereo matching using fast correlation and dynamic programming techniques. 1999. Submitted.