Multi-Resolution Rectangular Subregioning Stereo Matching Using Fast Correlation and Dynamic Programming Techniques

by

Changming Sun

CSIRO Mathematical and Information Sciences
Macquarie University Campus
Locked Bag 17, North Ryde NSW 2113

CMIS Report No. 98/246

December 1998
Abstract

Stereo matching is important in the area of computer vision and photogrammetry. This paper presents a fast and reliable stereo matching algorithm which produces a dense disparity map by using a pyramid structure, fast cross-correlation, rectangular subregioning and dynamic programming techniques. Fast correlation is achieved by using the box filtering technique which 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 required is reduced. The disparity for each scan line is found in the correlation matrix by finding the best path using dynamic programming rather than simply choosing the position that gives the maximum correlation coefficient. 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.

Keywords: Image matching, Stereo vision, Pyramid, Coarse-to-fine, Fast cross-correlation, Dynamic programming, Box filtering, Similarity measure, Rectangular subimages.
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 position of the scene points given knowledge about the relative orientation of the two cameras.

Because of factors such as noise, occlusion and perspective distortion, the appearances of the corresponding points will differ in the two images. For a particular feature in one image, there are usually several matching candidates in the other image. It is usually necessary to use additional information or constraints to assist in obtaining the correct match. Some of the commonly used constraints are:

1. Epipolar constraint: Under this constraint, the matching points must lie on the corresponding epipolar lines of the two images;
2. Uniqueness constraint: Matching should be unique between the two images;
3. Disparity gradient constraint: For certain kinds of 3D surfaces, the disparity gradient should be within a certain limit.

Matching techniques can be divided mainly into area-based, feature-based image matching, or a combination of them. Area-based methods have been applied successfully to aerial images, where the surfaces vary smoothly [1]. They have the advantage of directly generating dense disparity map but they tend to breakdown where there is lack of texture or where depth discontinuities occur [2]. The feature-based approaches match more abstract features, rather than matching texture regions in the two images [3, 4, 5]. Feature-based method provides more precise positioning for the matching result. It is also more reliable than area-based matching. Because of the sparse and irregularly distributed nature of the features, the matching result must be augmented by an interpolation step if a dense map of the scene is desired. If a feature-based method is used, an extra stage is needed for feature detection in the two images, which will increase the computational cost.

Lotti and Giraudon [6, 7] used a correlation based algorithm with an adaptive window-size that is constrained by an edge map extracted from the image. They presented results on real aerial images. 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. They also used ground control points to eliminate sensitivity to occlusion cost. Xiong et al [9] presented a stereo matching approach which integrates area-based and feature-based processes. Fua [10] described a correlation based multi-resolution algorithm which is followed by interpolation. Anandan [11] described a hierarchical computational framework for the determination of dense motion fields from a pair of images. A number of researchers have used dynamic programming to solve globally the matching problem [12, 13, 14, 15, 16].

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 [17, 18], and their performance and computation cost vary. The most commonly used similarity measure is the cross-correlation coefficient. It is popular because it corresponds to optimal signal-to-noise ratio estimation [19]. The sum of absolute differences (SAD) and the sum of square differences (SSD), both dissimilarity measure, have also been used. Their usage is usually justified on the ground that they are easy to implement and use less computing power, especially, when they are used in the fast sequential similarity detection algorithm [20, 21]. Barnea and Silverman [21] introduced a class of sequential algorithms for fast image registration. They were designed to reduce computation in matching procedures using minimum dissimilarity measures like the sum of the absolute differences (SAD). Konecný and Pape [22] reviewed image correlation techniques according to photogrammetric and mathematical fundamentals.

It has also been shown that the zero-mean normalized cross-correlation (ZNCC) and the zero-mean sum of squared differences tend to give better results [23, 24, 17, 18]. We will use the zero-mean normalized cross-correlation coefficient as the similarity measure of the candidate matching areas. The estimate is independent of differences in brightness and contrast due to the normalization with respect to mean and standard deviation. But the direct calculation of ZNCC is computationally expensive compared with SAD or SSD.

In this paper we address some of the efficient and reliable implementation aspects of the stereo matching algorithms by using rectangular subregioning, fast correlation and dynamic programming techniques in a multi-resolution scheme, which results in very fast stereo matching. The rest of the paper is organised as follows: Section 2 describes the box filtering techniques and derives our fast correlation method. Section 4 proposes the rectangular subregion method over the input images to further reduce the computation cost and memory requirement. The detailed matching method is described in Section 5. Section 6 shows the experimental results obtained using our fast stereo matching method applied to a variety of images. Section 7 discusses the reliability and computation speed issue of our algorithm. Section 8 gives concluding remarks.

2 Fast Cross-Correlation

Faugeras et al [24] developed a recursive technique to calculate the correlation coefficients which are invariant to the correlation window size. In this section, we approach the issue by using the Box-filtering technique.

Let \( f_{mn} \) 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 normalized cross-correlation of two windows can be written as follows:

\[
c_{ij,d} = \frac{\text{cov}_{ij,d}(f, g)}{\text{var}_{ij}(f) \times \text{var}_{ij,d}(g)}
\]
where

\[ \text{cov}_{ij,d}(f, g) = \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} (f_{m,n} - \bar{f})(g_{m+d,n} - \bar{g}) \]  

(2)

\[ \text{var}_{ij}^2(f) = \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} (f_{m,n} - \bar{f})^2 \]  

(3)

\[ \text{var}_{ij,d}^2(g) = \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} (g_{m+d,n} - \bar{g})^2 \]  

(4)

and \( d \) is the shift of the window along epipolar lines, which indicates the disparity; \( K \) and \( L \) define the correlation window size. \( \bar{f} \) and \( \bar{g} \) are the mean values within the windows. It can be seen from this equation that the co-variance between \( f \) and \( g \) and the variances of \( f \) and \( g \) at different positions in the image need to be evaluated. From Eqs. (2)-(4) it can be seen that to achieve fast calculation of Eq. (1), one needs to have fast ways to obtain the mean and variance of a window and correlation values of two windows. The following subsections will describe how to achieve these.

2.1 The Box Filtering Technique

McDonnell [25] described a box-filtering procedure for mean calculation. The main advantage of box filtering is its speed, which approaches four operations for each output pixel and is independent of box size. A brief description of the technique is as follows (the detailed description can be found in [25]). A \((2K+1) \times (2L+1)\) box filter is simply:

\[ \bar{f}_{ij} = \frac{1}{(2K+1)(2L+1)} \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} f_{mn} \]  

(5)

where the denominator is a constant. Each output row of \( \bar{f} \) is calculated using the window \( ABCD \) in \( f \) (referring to Fig. 1). A buffer \( IBUF(N) \) is maintained for this window. Each element of \( IBUF \) is the sum of the pixels in the corresponding column of the window. That is

\[ IBUF(j) = \sum_{m=i-K}^{i+K} f_{mj} \]  

(6)

After each row of \( \bar{f} \) has been calculated, the window moves down one row, and \( IBUF \) is updated by adding the new row and subtracting the old one as indicated in Fig. 1.

Each \( \bar{f}_{ij} \) is calculated using the box \( EFGH \). A value \( ISUM \) is stored for each box position given by

\[ ISUM = \sum_{n=j-L}^{j+L} IBUF(j) \]  

(7)

As the box moves, \( ISUM \) is updated by adding in the new \( IBUF \) value and subtracting out the old. Thus

\[ \bar{f}_{ij} = \frac{ISUM}{(2K+1)(2L+1)} \]  

(8)

When the first row of \( \bar{f} \) is calculated, \( IBUF \) must be initialized explicitly. Similar
initialization must be performed for the first value of each row of ISUM.

Fig. 1: The windowing process of the box filter operation.

2.2 Fast Calculation of Variance

Rearranging Eq. (3), the following equation (Eq. (9)) is obtained. It can be seen that the pixel variance within the box can also be obtained quickly during the same pass as when calculating the mean as described in subsection 2.1. This is achieved by accumulating the square of the intensity values while accumulating original pixel values for mean calculation. The first term \( \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} f_{m,n}^2 \) at the r.h.s. of Eq. (9) can be obtained during the same pass as calculating \( \bar{f} \). The variance of points within the box is calculated using Eq. (9).

\[
\begin{align*}
\text{var}_{ij}^2(f) &= \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} (f_{m,n} - \bar{f})^2 \\
&= \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} (f_{m,n}^2 - 2 \times f_{m,n} \times \bar{f} + \bar{f}^2) \\
&= \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} f_{m,n}^2 - (2K + 1)(2L + 1)f^2
\end{align*}
\]

Therefore we have a fast way to obtain the mean and variance of the input images for the calculation of the cross-correlation that is to be used as a reliable measure of similarity between matching candidates from the left and right images.

2.3 Fast Cross-Correlation

Here again we will use the technique described in Section 2.1 to achieve fast calculation of the cross-correlation term. Rewriting Eq. (2), we have:
\[
\text{cov}_{ij,d}(f, g) = \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} (f_{m,n} - \bar{f}) \times (g_{m+d,n} - \bar{g}+d)
\]

\[
= \sum_{m=i-K}^{i+K} \sum_{n=j-L}^{j+L} (f_{m,n} \times g_{m+d,n} - \bar{f} \times g_{m+d,n} - f_{m,n} \times \bar{g}+d + \bar{f} \times \bar{g}+d)
\]

Eq. (10) is the numerator of Eq. (1). Most, if not all, of the image correlation in the literature is performed using direct calculation of Eq. (10). Direct calculation of Eq. (10) has \((2K + 1)(2L + 1)\) redundancies. Similar to the fast calculation for the mean and variance, cross-correlation of two images can be obtained using only a few multiplications. The first term of the r.h.s. of Eq. (10) is the summation of the pixel multiplications over the correlation window with the right image shifted \(d\) pixels. This operation can also be performed using the same Box-filtering idea to achieve fast computation speed. Similar to the process of calculating the variance as described in the previous subsection, the multiplication of \(f_{m,n}, g_{m+d,n}\) is used rather than \(f^2_{m,n}\). The second term of the r.h.s. of Eq. (10) is straight-forward calculation using the available mean values.

In our case, the correlation of two windows in the two images is performed along the same horizontal scan line. If for any point in the left image, the search window is assumed to be within \([-w, +w]\) in the right image, then the value of \(d\) in Eq. (10) varies from \(-w\) to \(+w\). The traditional way of obtaining the correlation is to fix a point in the left image and vary \(d\) within \([-w, +w]\) in the right image to calculate the correlation coefficients.

In our new algorithm for fast correlation, we first fix on one particular \(d\) all the points in the left image and calculate the cross-correlation between the whole left image and the whole shifted right image of the amount \(d\). After this, for every point on the left image we have a 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 each of these plane together we have a correlation cube.

The complexity of the algorithm is \(O(MND)\), where \(M, N\) are the image row and column numbers and \(D\) is the maximum disparity search range. \(D\) is obtained in each pyramid level by searching the minimum and maximum disparities. The storage space needed for the correlation coefficients is in the order of \(4MND\) bytes.

### 2.4 Correlation Cube

The result of the correlation calculation as described in Section 2.3 is a cube containing the correlation coefficients as shown in Fig. 2. The size of the cube depends upon the image size \((MN)\) and the disparity range \((2w + 1)\).
Fig. 3 shows one horizontal slice of the correlation cube. Fig. 3(a) is the correlation performed using left image as reference and right image as target image; and (b) illustrates the correlation performed using right image as reference and left image as target image. The shaded region in the slice contains no correlation value, i.e. there is no window overlap for correlation. Fig. 3(a) and (b) show the symmetric role between the left and right images. Each slice of this correlation cube will be used later to obtain disparity values.

Fig. 2: An illustration of the correlation cube obtained after using the fast correlation method.

Fig. 3: One horizontal slice of the correlation matrix. (a) Correlation performed using left image as reference and right image as target image; and (b) Correlation performed using right image as reference and left image as target image.

3 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 [26], 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 [27]: (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.

Fig. 4 shows the process of building the image pyramids. In the current implementation, the lower resolution image is obtained by simply taking the average value of the corresponding $r \times r$ pixels in the previous higher resolution level for its simplicity.

![Image pyramids construction](image)

During the process of projecting the disparity map from the current level of the pyramid to the next (if current level is not level 0, or the highest image resolution), the disparity image size was scaled up by the value of $r$, where $r$ is the reduction ratio used when building the image pyramid, and the disparity value was scaled up by the same $r$. The disparity value where the position $(i, j)$ of the next level image is not a multiple of $r$ was obtained by bilinear interpolation.

4 Work with Subimages by Rectangular Subregioning

Rather than work with the whole image during the fast image correlation stage as described in the previous sections, 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, i.e. every pixel is shifted with the same value and in the same disparity search range, where $M$, $N$ are the image row and column numbers (they do not have to be the power of two) and $D$ is the disparity range over the whole image. $D$ is the difference between the maximum disparity and the minimum disparity.

If the image is divided into $n$ subimages or rectangular subregions as will be described below, 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 $i$th subimage or region, and $D_i$ is the disparity 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.
Although there are some overheads when working with subimages, such as region segmentation and house-keeping, the time saved during the correlation stage is far greater than the time spent for the overhead. There is another advantage for working with subimages in terms of memory usage. As mentioned in section 2.4, 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 $\max_i(4M_iN_iD_i)$, because the memory for each subregion is dynamically allocated and freed.

![Fig. 5: Sub-dividing the whole image into subimages. Fast correlation is performed on each of the rectangular subregions.](image)

![Fig. 6: An example result of sub-dividing the whole image into subimages based on intermediate disparity map in the pyramid. (a) a disparity map at a particular image pyramid; (b) the disparity map shown in (a) overlaid with the rectangles obtained. Each of these rectangles will be used for running the fast correlation algorithm described earlier.](image)

Now we will describe our method for segmenting an image into rectangular subregions. Because the shapes of the search window and the correlation image regions are all rectangular, the regions of subimages need to be rectangular. Also because that we work on the epipolar images and we perform dynamic programming on a correlation matrix for each scan-line to find the disparity, we divide the input image into several horizontal stripes first. Then for each horizontal stripe, several vertical cuts are conducted. The resultant segmentation of the input image is somewhat in the form as shown in Fig. 5. For example, the segmented image contains horizontal stripe $AA'B'B'$, and this stripe is then cut into smaller rectangles. One of these smaller rectangles is $PQST$. Fast
correlation is performed on each of these smaller rectangle images, and the obtained correlation coefficients are put together into horizontal stripes or cubes.

The method that we developed for segmenting an image into rectangles are in the line of region merging techniques. The input image is first divided into thin horizontal stripes. The input for this segmentation step is the intermediate disparity image by projecting and interpolating the result from the previous pyramid levels. If the current level is at the top of the pyramid, the current disparity image is set to zero. The coordinate of the disparity image is the same as the left image if the left image is take as the reference; otherwise, the coordinate of the disparity image is the same as that of the right image. Each stripe contains the property such as stripe corner positions, the minimum and maximum disparities. Then these thin horizontal stripes are merged according the the criteria that the overall computing complexity is minimum by taking the overhead into account. At each step of the merging process, only neighbouring stripes can be merged.

After the image has been segmented into horizontal stripes, each such stripe can then be cut into regions by vertical lines. The steps are similar to those for segmenting images into horizontal stripes. The objective is to obtain large regions with small disparity range and small regions with large disparity range. The grey region $PQST$ in Fig. 5 illustrates one of the subregions. Fig. 6(b) shows the subregions obtained from Fig. 6(a). When actually performing fast correlation calculation for each of the subregions, certain size of region overlapping as shown in Fig. 7 by the small dotted region 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. In the case when the left image is taken as the reference image, the subregions obtained on the disparity map also correspond to the subregions in the left image. Otherwise, the subregions obtained on the disparity map will correspond to the subregions in the right image. 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.

The approach we used here for rectangular subregioning may not be the global
minimum, but it is fast and simple and serves our purpose for fast processing.

5 Matching Strategy

5.1 Best Path in the Matrix

Most researchers [2] choose the position that gives the maximum correlation coefficient as the disparity value for any point in the left image. We choose a slice of the correlation coefficient cube as a 2D correlation matrix for each scan line of the input image and use this matrix to obtain more reliable disparities. The width of the matrix is the same as the length of the horizontal image scan line, and the height of the matrix equals the correlation search range, \(2w+1\). A typical correlation matrix is shown in Fig. 3(a). This matrix is actually one slice of the correlation cube obtained in Section 2.3. We will use the correlation matrix to find the disparity for any one scan line. Rather than choosing the maximum correlation coefficient, we find a best path from left to right through the correlation matrix. The position of the path indicates the best disparity for this scan line.

![Correlation matrix and path](image)

Fig. 8: One horizontal slice of the correlation matrix for the images shown in Figure 13(a)(b). (a) Correlation matrix obtained by using left image as reference and right image as target image; and (b) The path obtained in the correlation coefficient matrix using dynamic programming.

The algorithm for finding the best path through the correlation matrix is performed by using a dynamic programming technique [28, 13, 12]. Fig. 8 shows an example of the path obtained using dynamic programming techniques. Fig. 8(a) gives a correlation matrix for the images in Figure 13(a)(b). Fig. 8(b) is the path obtained in the correlation coefficient matrix using dynamic programming technique. The best path gives the maximum of summation of the correlation coefficients along the path when certain constraints are imposed. The disparity gradient limit constraint can be easily implemented during the dynamic programming minimization process. This limit constrains the size of neighbour search or the path that it can go.

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 position of this sub-pixel can be found at 
\[ x = -\frac{b}{2c} \]
as illustrated in Figure 9. If only three points of the correlation values are used, e.g. the points \( i - 1, i, i + 1 \), the sub-pixel position of the disparity can be calculated using the following formula [29]:

\[
x = i + \frac{1}{2} \times \frac{C(i - 1) - C(i + 1)}{C(i - 1) - 2C(i) + C(i + 1)} \tag{11}
\]
where \( C(i) \) is the correlation value in the matrix at position \( i \), and \( x \) is the sub-pixel disparity position obtained.

![Fig. 9: Sub-pixel disparity accuracy by finding the peak of a fitted quadratic parabola equation.](image)

5.2 Occlusion Detection

Using the fact that the roles of the left and right images are symmetric, we calculate the disparities from left to right images and from right to left images. For most of the areas in the image, the two disparities obtained will differ only in sign. In the regions when the magnitude of disparities are also different, it is likely that the region contains occlusion or mismatch has occurred. In this case certain procedure needs to be carried out to remedy this. We treat these regions as un-defined and perform interpolation using neighbouring disparity values.

5.3 Non-overlapping Region

For the pair of images to be matched, there are almost always some non-overlapping regions. In these regions the object only appears in one of the image and not the other. During the image matching process, nonsense results usually arise in these regions. Therefore it is necessary to identify those regions and take appropriate actions. The non-overlapping region can be detected using the correlation coefficient values. If the correlation values of a region (which is close to the image boundary) is less than a threshold, this region can be treated as a non-overlapping region. Fig. 10 gives an
example image showing the correlation coefficient values at each position of the original image shown in Fig. 13(a)(b). The dark regions, which have small correlation values, show that the correlation between the left and right images are small. For those regions with low correlation values and close to the image boundaries, they are likely to be the non-overlapping regions. The black stripe close to the left boundary is the region which does not have matching points in the right image. The black stripe close to the bottom boundary of the image is a region where there is no texture, and therefore cannot be matched.

Fig. 10: The correlation values for the images shown in Figure 13(a)(b).

5.4 Algorithm Steps

Our proposed algorithm for stereo matching is:

1. Build image pyramids with $k$ levels (from 0 to $k - 1$), with the reduction ratio of $r$, from the original left and right images; The upper or coarse resolution levels are obtained by averaging the corresponding $r \times r$ pixels in the lower or finer resolution level as shown in Fig. 4;
2. Initialize the disparity map as zero for level $k - 1$ and start stereo matching at this level;
3. Perform image matching using the method described in Sections 2-5 which includes:
   (a) Segment images into rectangular subregions;
   (b) Perform fast zero-mean normalised correlation to obtain the correlation coefficients;
   (c) Use dynamic programming to find the best path, which will then give the disparity map.
4. If $k \neq 0$, propagate the disparity map to the next level using bilinear interpolation, set $k = k - 1$ and then go back to step 3; otherwise go to step 5;
5. Display disparity map.
6 Experiment Results

This section shows some of the results obtained using our method described in this paper. A variety of images have been tested, including Random Dot Stereograms (RDS), synthetic images, and different types of real images. The image size does not have to be a power of 2. The input left and right images are assumed to be rectified epipolar images.

A. Random Dot Stereograms

Fig. 11 shows the results obtained by applying the algorithm to Random Dot Stereograms. Fig. 11(a)(b) show the original left and right RDS of a “wedding cake”. The input image size is 141 by 151. Fig. 11(c) is the disparity map obtained showing the different levels of “cakes”. Fig. 11 (d)(e) show the original left and right RDS of a different object, and Fig. 11(f) is the disparity map obtained showing the “cube”.

B. Synthetic Images

Fig. 12 gives the result of the algorithm running on two pairs of synthetic images. Shown in the top row are the two images of two different depths of a background. The bottom row of the same figure shows a concrete sphere on a table. The sizes of both of these images are 256×256. The left hand side of Fig. 12(c)(f) contains a stripe of white or black regions which indicates that the same regions in the left image does not have corresponding pixels in the right image.

C. Real Images (close-range images and aerial photos)

Fig. 13(a)(b) show the original left and right images of a baseball on a newspaper. Fig. 13(c) is the disparity map obtained, and (d) gives a perspective view of the disparity map. Fig. 13(e) shows the occluded regions obtained. Fig. 14 shows the results for five close-range real images. Fig. 15 shows the results for four aerial photo images. Many other different 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.

D. Images from Outer Space

Fig. 16(a)(b) show a pair of images of “Big Crater” taken by the Mars Global Surveyor spacecraft of Malin Space Science Systems/NASA. Fig. 16(c) is the disparity map recovered. Fig. 16(d)(e) are the stereo images of a rock from the Mars Pathfinder Images. Fig. 16(f) gives the recovered disparity map.

E. SEM Images

A number of scanning electron microscopic (SEM) images were also used for testing our algorithms. The top row of Fig. 17 shows the pair of images of “flower-like” crystals and the disparity map obtained. The bottom row of Fig. 17 gives the two SEM images of cement air block and the disparity map obtained. Note that in the original images there are two regions that are very dark.

F. Running Times

The computer used is a 85MHz Sun SPARCserver1000 running Solaris 2.5. The
Fig. 11: The matching result for Random Dot Stereograms. (a,d) left image; (b,e) right image; and (c,f) the disparity map recovered. (Images (d,e) courtesy of Bill Hoff at the University of Illinois [30].)

Fig. 12: The matching result for synthetic images. The image sizes are 256×256. Top row shows a two-level background. Bottom row are the images of a sphere on a table. (a,d) left image; (b,e) right image; and (c,f) the disparity map recovered. (Images (a,b,d,e) courtesy of Bill Hoff at the University of Illinois [30].)
typical running time for the algorithm on a 256×256 image is in the order of seconds rather than minutes or even hours. Table 1 gives some of the typical running times of the algorithm on different size of images with different disparities. The size of the correlation window used for the images shown in the table is 9×9. The reduction ratio $r$ used in the pyramid generation process is 2. The maximum disparity gradient used is 1. The time shown in the table includes the pyramid building process. For example, for the “ball” image of size 256×256 as shown in Fig. 13(a)(b), the program only takes 4.95 seconds. It only takes 19.60 seconds to obtain the disparity for the “pentagon” image.

The time shown for “User time1” is obtained without using the subimages method as described in Section 4, while the time shown for “User time2” is obtained by using the subimages method. It can be seen that the time spent by the algorithm using subimages method is almost half of the time without using the subimage method. Interested people could try their own images by accessing the Web page given in Section 9.

7 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. As mentioned in Section 3, the upper levels of the pyramids are ideal to get an overview of the image scene. Therefore the matching in the upper levels will have a more global effect. The details can be found down the pyramid at higher resolution. (2) The zero-mean normalized cross-correlation similarity measure is used, which is independent of differences in brightness and contrast due to the normalization with respect to mean and standard deviation, rather than using the relatively computationally cheap measure using SAD or SSD.
Fig. 14: The matching result for some close-range images. (a,d,g,j,m) left image; (b,e,h,k,n) right image; and (c,f,i,l,o) the disparity map recovered. (Images (a,b,d,e,g,h,j,k) courtesy of Bill Hoff at the University of Illinois [30]. Images (m,n) courtesy of Gerard Medioni at USC Institute for Robotics and Intelligent Systems [3].)
Fig. 15: The matching result for some aerial photo images. (a,d,g,j) left image; (b,e,h,k) right image; and (c,f,i,l) the disparity map recovered. (Images (a,b) courtesy of Bill Hoff at the University of Illinois [30]. Images (d,e,g,h,j,k) courtesy of Institute of Photogrammetry, Univ. of Stuttgart, Germany.)
Fig. 16: The matching result for two pairs of images from space. (a,d) left images; (b,e) right images; and (c,f) the disparity maps recovered. (Images courtesy of Malin Space Science Systems/NASA.)

Fig. 17: The matching result for two pairs of SEM images. (a,d) left images; (b,e) right images; and (c,f) the disparity maps recovered. (Images courtesy of Philips Electron Optics, Eindhoven, The Netherlands.)
Table 1: Running times of the algorithm on different images. The size of the correlation window is 9×9. The reduction ratio r used in the pyramid generation process is 2. The maximum disparity gradient used is 1. The ball image is shown in Fig. 13(a)/(b). The pentagon image is shown in Fig. 15(a)/(b). The circuit image is shown in Fig. 14(d)/(e). The flat image is shown in Fig. 15(d)/(e).

<table>
<thead>
<tr>
<th>Image name</th>
<th>Image size</th>
<th>Pyramid levels</th>
<th>Search range</th>
<th>Disparity range</th>
<th>User time1</th>
<th>User time2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ball</td>
<td>256×256</td>
<td>2</td>
<td>[-9,9]</td>
<td>[-19,7]</td>
<td>7.60s</td>
<td>4.95s</td>
</tr>
<tr>
<td>pentagon</td>
<td>512×512</td>
<td>3</td>
<td>[-2,2]</td>
<td>[-13,12]</td>
<td>30.05s</td>
<td>19.60s</td>
</tr>
<tr>
<td>circuit</td>
<td>512×512</td>
<td>3</td>
<td>[-5,5]</td>
<td>[-27,29]</td>
<td>42.67s</td>
<td>21.98s</td>
</tr>
<tr>
<td>flat</td>
<td>1000×1000</td>
<td>4</td>
<td>[-3,3]</td>
<td>[-39,27]</td>
<td>172.14s</td>
<td>98.75s</td>
</tr>
</tbody>
</table>

(3) The correlation coefficient matrix is used as input to the dynamic programming stage. A number of approaches that use dynamic programming method just use the intensity value along the left and right epipolar lines. These approaches do not take the neighbourhood information from the successive scan-lines into account. Some researcher do take the inter-scanline information into account. In our approach, by using the correlation coefficient value which is obtained from a local window during the cross-correlation step, information from different scan-lines have been used. (4) Dynamic programming technique is used to find a path in the correlation matrix. Given the correlation matrix which corresponds to the correlation values within the search range for each point on the left image, one may search for the position which gives maximum value as the match point. This may have the effect that certain isolated points may be generated. By using the dynamic programming technique on the input correlation coefficient matrix, one will obtain a more smooth path within the matrix. As most of the other correlation based matching methods, if there is a large area with little texture, therefore the correlation coefficients for a number of points in this area may be undefined, the matching result may not be very pleasing. However, if this area is not too large, the effect of using the dynamic programming technique to find a path will have the effect of filling the holes with undefined correlation values.

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 normalized cross correlation is developed. The original idea of box-filtering for calculating image mean was developed further for fast calculation of image variance at the same time when one calculates the image mean. The fast cross-correlation between two images are achieved by fixing one shift for every points on the left image and calculating the cross-correlation in the way similar to that when one calculates the image variance. This way the redundant computation is eliminated and fast computation is achieved. (2) We have developed a new rectangular subregioning technique for fast computation of correlation coefficients. Rather than working with the whole image when perform cross-correlation, the input images are sub-divided into rectangular subregions depending on the current disparity map in a certain level of the pyramid. These regions tend to have the property that when the disparity range is small the size of the region is large, and when the disparity range is large the size of the
region is small. The end effect of these are the reduced computation cost. (3) Apart from having the advantages of increasing the reliability, the coarse-to-fine approach is also faster than one without using it. The smaller image size on the upper levels of the pyramid gives a rough estimate of the disparity. The larger image size on the lower levels of the pyramid can use the initial estimate from the previous level to refine the disparity value in a reduced search range.

8 Conclusions

We have developed a fast and reliable stereo matching method using rectangular sub-regioning, fast correlation and dynamic programming techniques in the coarse-to-fine framework. The algorithm produces a reliable dense disparity map: for each point in the image, a disparity value is obtained. It also works up to the image boundary. The fast cross-correlation method was developed from the box-filtering idea. The time spent in the stage for obtaining the mean and standard deviation for the normalized 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 fast correlation of two subregions from the left and right images is achieved by shifting the whole subregion. The sub-windowing technique was also used 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 implemented on standard computers, and no special hardware is used. The algorithm could be implemented in parallel since the processing in different parts of horizontal scanlines are independent of each other.

By using the zero-mean normalized cross-correlation (ZNCC) similarity measure rather than the simple SSD or SAD, together with the multi-resolution scheme and dynamic programming techniques, the reliability of the algorithm was increased.

The disparity gradient constraint was applied during the search of disparity between left and right scan-lines by using dynamic programming. Occlusion can be detected using the symmetric role of the left and right images. Non-overlapping regions of the input images can also be detected using the correlation coefficients obtained.

The algorithm was shown to be fast and reliable by testing on several different types of images: both synthetic and real images.

9 WWW Demo

There is a WWW page setup to allow interested people to run our algorithm using their own stereo images. The WWW demo address is at:

http://extra.cmis.csiro.au/IA/changs/stereo/
Acknowledgement

The author is grateful to the author or owners of the images used in this paper. We thank the colleagues at CSIRO Mathematical and Information Sciences for many useful discussions and suggestions.

References


