Dense Stereo Correspondence Based On Recursive Adaptive Size Multi-Windowing

Sylvia Oi-Yee Chan, Ya-Ping Wong, Jacob K. Daniel
Faculty of Information Technology,
Multimedia University,
Jalan Multimedia, 63100 Cyberjaya,
Selangor Darul Ehsan, Malaysia
{sylvia.chan, ypwong, jacob.daniel}@mmu.edu.my

Abstract

An efficient technique of stereo matching to compute dense disparity map is presented. The technique is point-oriented and uses multiple windows to enhance the strategy of finding a best match. In addition to that, a novel technique for adapting the window size based on multiple windows is discussed. The window size is recursively adapted using normalized sum of squared differences similarity measure and also the output of the multi-windowing method. Several constraints like epipolar constraint, uniqueness constraint and mutual correspondence constraint are also incorporated into the algorithm to prune bad matches. The results of the technique which has been tested on real images are also presented.

Keywords: multi-windowing, adaptive window, mutual correspondence, dense stereo matching, sum of squared differences

1 Introduction

Stereo matching procedure has been widely used in many applications of view synthesis and 3-D reconstruction algorithms. One of the main concerns in these algorithms is the correspondence problem, where the main objective to solve this problem is to determine an item in the left image and an item in the right image that are projections of the same scene element. Earlier works of view synthesis and 3-D reconstruction require either dense correspondence [1,2] or sparse correspondence [3,4] between the source images.

However, we based our discussion in this paper on computing dense correspondence where a best match on the right image is obtained for every pixel on the left image. Because our objective is to obtain a dense correspondence map, we have used area-based method instead of feature-based method in the implementation. Area-based method for stereo correspondence uses small image windows typically centered at a given pixel and a similarity measure computed based on the pixels inside the window for matching assuming that the grey levels are similar.

In section 2, we will briefly explain the need for multi-windowing A novel approach to adaptive size multi-windowing based on normalized sum of squared differences is discussed in section 3. Implementation of several constraints will be covered in section 4. In section 5, we will present the overall algorithm.

2 Multiple windows

In this paper, we assume that conjugate pairs lie along raster lines and that the image intensity of a 3-D point is the same on the two images. For the matching procedure, we have implemented the normalized sum of squared differences (SSD) correlation algorithm as a similarity measure. The normalized SSD equation that we used is as stated below:

$$S(x,y,d)=\frac{\sum_n\sum_m\left[I_l(x+m,y+n)-I_r(x+m+d,y+n)\right]^2}{\sqrt{\sum_n\sum_m I_l(x+m,y+n)^2\cdot\sum_n\sum_m I_r(x+m+d,y+n)^2}} \quad (1)$$

where I_l and I_r are the left and right images respectively and $S(x,y,d)$ is the normalized sum of squared differences of the pixel (x,y) whose disparity is to be calculated using a $m\times n$ window. d is the offset in the right image with respect to the pixel (x,y) in the left image. The computed disparity is the one that has a selected window with a suitable size which minimizes equation (1).

Area-based matching that utilizes correlation window which covers a region of non-constant disparity is likely to fail. Therefore, correlation is performed using nine windows [5] as in figure 1 and the disparity with the smallest SSD error is retained for each pixel in the image. The window that yields a smaller SSD error is more likely to cover a constant depth region. In this way, the selection of an appropriate window is driven by the disparity profile itself.

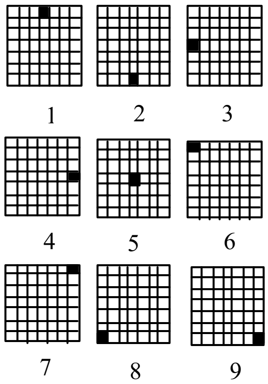
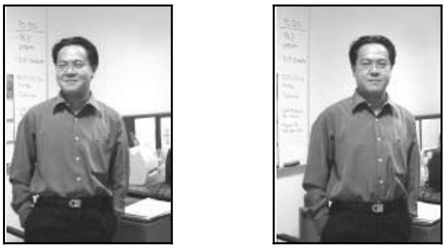


Figure 1: The nine asymmetric correlation windows. The pixel for which disparity is computed is highlighted.

Figure 2 shows the image pair that we used for testing and the results of performing matching on these images using a single window are shown in figure 3. The intensity varies inversely with depth of the object. Higher intensity pixels indicate lower depth (nearer objects). We see clearly that implementing the algorithm with multiple windows give better results as illustrated in figure 4.



(a) (b)

Figure 2: The (a) left and (b) right images used in our implementation.

(a) (b)

Figure 3: Matching performed from (a) left to right (b) right to left using a single 11X11 window.

(a) (b)

Figure 4: Matching performed from (a) left to right (b) right to left using multiple 11X11 windows.

3 Adaptive window size

In addition to selecting the appropriate window that covers a constant depth region, the window size also plays an important role in assisting the search for a best match. The window size must be large enough to include sufficient intensity variation for reliable matching, but small enough to avoid the effects of projective distortion [6]. In other words, increasing the size of the window will result in growing error of depth estimates but reducing its size will make the computed disparity more noise-sensitive. Figure 5 shows the results of matching using multiple windows of fixed size and implementing mutual correspondence constraint which will be explained in the following section. We observe that the fixed size windows outperform one over the other in certain areas of the image despite using multiple windows.

(a) (b)

Figure 5: Matching done using multiple windows of a fixed size (a) 11X11 (b) 5X5

After selecting the suitable window by following the technique in the previous section, we now reduce the size of the selected window recursively to get a window that will further minimize the SSD error. The reduction procedure will take place either until the SSD error is greater than the previous value or it reaches a minimum window size.

The window size is reduced from all direction towards the pixel for which disparity is computed excluding the side on which the pixel lies. For example, the first window in Figure 1 has the pixel of interest located in the center of the top row. Thus to reduce the size of this window, we simply take away a column from the left and right side and the bottom two rows of the window. By doing so, we have reduced the 7X7 window to a 5X5 window. See figure 6. Table 1 summarizes the dimension reduction procedure for all nine windows.

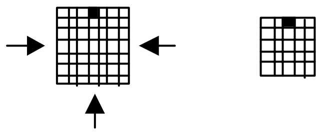


Figure 6: The 7X7 window which the pixel's disparity is to be calculated is located in the center of the top row is reduced in size to a 5X5 window.

Table 1: Summary of dimension reduction based on multiple windows.

Window	Number of rows/columns to be deleted on the following borders			
	Left	Right	Top	Bottom
1	1	1	0	2
2	1	1	2	0
3	0	2	1	1
4	2	0	1	1
5	1	1	1	1
6	0	2	0	2
7	2	0	0	2
8	0	2	2	0
9	2	0	2	0

4 Implementation of constraints

The inherent ambiguity of the correspondence problem can in practical cases be minimized using several constraints [7]. Among the constraints that we have applied are the epipolar constraint, uniqueness constraint and the mutual correspondence constraint.

The epipolar constraint says that the corresponding point can only lie on the epipolar line in the second image. This results in the search being carried out in 1-D. In this paper, it was assumed that the epipolar lines are aligned and horizontal so that we perform the search along corresponding x-axis.

The uniqueness constraint limits the pixel on the left image to correspond to at most one pixel on the right image and is also implemented in our algorithm.

We also enforced mutual correspondence constraint that helps to rule out points that do not have a corresponding counterpart due to occlusion, highlight, or noise. Assume the search started from the left image point p_L and a corresponding p_R was found. If the task is reversed, and a search starting from the point p_R fails to find the point p_L, then the match is not reliable and should be ruled out.

5 The adaptive size multi-windowing algorithm

We now present the complete algorithm. As mentioned earlier, we are implementing the area-based correlation algorithm and therefore the procedure is performed on every pixel (x,y) on the left image, I_L. For each of these pixels, a best match based on the minimum SSD error is searched upon the right image, I_R , for all possible disparities, d. d ranges from the minimum to the maximum possible disparities specified by the user. For each of these disparities in turn, the SSD error is calculated using all the nine windows in figure 1 and the one that gives the minimum error is retained.

After that, the selected window, its decreased size, *size-2*, and the minimum error it gives are passed to the recursive function RWS. When all the loops terminate, we check for mutual correspondence. If they do not mutually correspond as described in section 4, we mark the pixel as occluded. Refer to figure 7.

```
for all (x,y) in I_L do
    for all d do
        for all w do
            d_L,w(x) = arg min_d S(x,y,d; I_L, I_R, w)
            d_R,w(x) = arg min_d S(x,y,d; I_R, I_L, w)
        end for
        d_L,w(x) = RWS(size-2,
                   w, S(x,y,d_L,w(x)), true)
        d_R,w(x) = RWS(size-2,
                   w, S(x,y,d_R,w(x)), false)
    end for
end for

for all (x,y) in I_L do
    if (d_L(x) ≠ -d_R(x+d_L(x))) then
        mark pixel as occluded
    end if
end for
```

Figure 7: Overall of the adaptive size multi-windowing algorithm

The method RWS is a recursive function that decreases the window size of the appropriate window and computes the new SSD error. See figure 8. This function returns when the SSD error computed is larger then the previous value or when the minimum window size has been reached. *leftright* is a flag that indicates whether matching is performed from left to right or otherwise. Figure 9 shows the result of the presented algorithm.

```
int RWS(currentSize, w, previousSSD, leftright)
{
    if (leftright=true) then
        d(x) = arg min_d S(x,y,d, I_L, I_R, w)
    else
        d(x) = arg min_d S(x,y,d, I_R, I_L, w)
    end if
    if (currentSize>minimum window size) then
        if (S(x,y,d(x))<previousSSD) then
            d(x) = RWS(currentSize-2,
                       w,S(x,y,d(x)),leftright)
        else
            return d(x)
        end if
    else
        return d(x)
    end if
}
```

Figure 8: The recursive RWS function

Figure 9: Result of matching using adaptive size multiple windows.

6 Conclusion

The algorithm presented in this paper harnesses the advantages of using multiple windows and adaptive size windows for stereo matching. This technique was based on the normalized SSD using the intensity values of the images. For future work, we may adapt this algorithm using chromatic information. Colour provides useful information for image matching and has been demonstrated in [8]. We have used the point-oriented strategy in our method where for each location in one image, we found the displacement that aligns this location with the best matching location in the other image. Thus, displacement-oriented scheme is also possible for future extension [9]. This scheme finds all the locations that match well given a certain displacement.

7 References

[1] Faugeras, 0. and Robert, L., "What can two images tell us about a third one?", *Proceedings of the Third European Conference on Computer Vision*, pp 485-492 (1994).

[2] Avidan, S. and Shashua,, A., "Novel view synthesis in tensor space", *In Conference on Computer Vision and Pattern Recognition*, pp 1034-1040 (1997).

[3] Chan, S.O.Y. and Wong, Y.P., "Automating 3-D model reconstruction algorithm for image-based modelling and rendering", *Proceedings of the Fourth European Workshop on Image Analysis for Multimedia Interactive Services* (2003).

[4] Chan, S.O.Y. and Wong, Y.P., "View synthesis of uncalibrated stereo images using feature-based method and singular value decomposition", *Accepted and to be published in Proceedings of the Third International Conference of Vision, Imaging and Image Processing 2003* (2003).

[5] Fusiello, A., Roberto, V. and Trucco, E., "Efficient stereo with multiple windowing", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp 858-863 (1997).

[6] Kanade, T., and Okutomi, M., "A stereo matching algorithm with an adaptive window: Theory and experiment", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 9 (1994).

[7] Sonka, M., Hlavac, V. and Boyle, R., *Image Processing, Analysis, and Machine Vision*, 2nd Edition, PWS Publishing, USA (1999).

[8] Koschan, A., "Using perceptual attributes to obtain dense depth maps", *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, pp 155-159 (1994).

[9] Scharstein, D., *View Synthesis Using Stereo Vision*, Lecture Notes in Computer Science (LNCS), Springer Verlag, vol. 1583 (1999).