Disparity-Based View Interpolation for Multiple Perspective Stereoscopic Displays

Ho-Chao Huang [a], Ching-Che Kao [b], Yung-Chieh Lin [b], Yi-Ping Hung [a,b]

[a] Institute of Information Science, Academia Sinica, Taipei, Taiwan
[b] Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan

ABSTRACT

This paper presents a new technique for generating multiview video from a two-view video sequence. For each stereo frame in the two-view video sequence, our system first estimates the corresponding point of each pixel by template matching, and then constructs the disparity maps required for view interpolation. To generate accurate disparity maps, we use adaptive-template matching, where the template size depends on local variation of image intensity and the knowledge of object boundary (which may be available from user specification or from some computer vision algorithms). Then, both the disparity maps and the original stereo videos are compressed to reduce the storage size and the transfer time. Based on the disparity, our system can generate, in real time, a stereo video of desired perspective view by interpolation or extrapolation from the original views, in response to the head movement of the user. Compared to the traditional method of capturing multiple perspective video directly, the approach of view interpolation can eliminate the problems caused by the requirement of synchronizing multiple video inputs and the large amount of video data needed to be stored and transferred.

Keywords: Stereoscopic Display, View Interpolation, Morphing, Disparity Estimation

1. INTRODUCTION

For many immersive entertainment [1] and virtual reality systems, [2-4] providing stereo views is an important feature to let users feel as if they are participating, especially if the systems can provide multiple perspective stereo views. There are many new multiple view display systems, which provide the viewer with the appropriate monoscopic or stereoscopic view of a scene, depending on his position. As an example, an autostereoscopic display system may be realized by using several projectors, e.g., six, and a lenticular screen. However, it is not easy to capture multiple perspective videos because multiple perspective video capture devices are expensive and unpopular. Even if one can capture multiple perspective videos that are synchronized correctly using special hardware, it is not easy to store and manipulate those video sequences due to the large amount of image data.

Compared with multi-channel video capture devices, the stereo video camera (two-channel) is more popular and less expensive. Thus, to generate multiple perspective video from one stereo video captured by the more popular stereo cameras becomes an important problem. Furthermore, in order to improve the quality of the generated intermediate views, a reliable disparity field has to be estimated. In our system, we first obtain the initial two views using a stereo video camera, and then estimate the corresponding points of each pixel in the stereo video automatically with epipolar constraints and then generate two disparity maps for each image pair in the stereo video. We use one disparity map to record the estimated right-to-left correspondence and the other the left-to-right correspondence. The generation of disparity maps is a time consuming procedure, but it can be done off-line. From the original stereo video and the estimated disparity maps, our system can generate multiple perspective stereo views on-line by interpolation or extrapolation from the original stereo video, and can allow the user to change viewpoint within a certain range around the original two views. Both the difficulty of video processing and the amount of video data can be reduced significantly. A preliminary version of this system was described in Ref. 5.

The remainder of this paper is organized as follows. Section 2 briefly reviews the multiple perspective video display systems and some morphing techniques commonly used for view interpolation. Section 3 describes the multiple perspective video generation system, which includes prewarping, segmentation, disparity estimation, data compression, novel view

In *Stereoscopic Displays and Virtual Reality Systems VII*, John O. Merritt, Stephen A. Benton, Andrew J. Woods, Mark T. Bolas, Editors, Proceedings of SPIE Vol. 3957 (2000) • 0277-786X/00/$15.00

generation process, and postwarping. Finally, experimental results are presented in Section 4, and conclusions and future works are given in Section 5.

2. REVIEW

2.1. Multiple Perspective Video Display System

Several kinds of stereo display systems are commonly used in multimedia applications. For example, the polarized stereo video projecting system is widely used in theater to show stereo movies. The audiences must wear polariscopic glasses to enjoy the stereo movie. In computer display systems, LCD shutter glasses is commonly used to see stereo objects on computer monitor. The blue-red stereo glasses is the least expensive device widely used for showing stereo objects on computer monitors, slide shows and printed articles. All the commonly used stereo display mechanisms can only display two perspective views at the same time, and the users need to wear some kind of stereo glasses to see the stereo output.

The multiple perspective video display system is an alternative method to show stereo videos. Fig. 1 shows an example of the multiple perspective video display system containing six video projectors. The six video projectors, numbered from one to six, project different perspective videos on the convex lens. The images of those videos converge at the front of the display system in reverse order. There are six possible positions for seeing the projected videos. When the user's eye is at position 1, the user can see the video projected by video projector 6, for example. If the user's eye is at position 2, he can then see video 5, and so on. Most of the multiple perspective video display systems are designed such that the user's left and right eyes are located at the adjacent viewing positions at the same time. For example, the user can put his left and right eyes at viewing position 3 and 4, respectively, to see videos 4 and 3 simultaneously to experience the stereo movie. The user can move his head to left or right to see different perspectives. For example, when the user's head moves to left, his left and right eyes move to viewing position 2 and 3, respectively, and the observed videos become video 4 and 5, which forms a stereo view of another perspective. Hence, the example display system shown in Fig. 1 can provide five stereo perspective views simultaneously, and the users need not to wear any kind of stereo glasses to see the stereo video.

It is obvious that the multiple perspective video display system is a better device for providing immersive feeling – at least, the object looks more real. However, it is not easy to obtain multiple perspective videos, and it is not easy to handle the large amount of multi-channel video data, especially when the number of channel is large. Our goal is to automatically generate multiple perspective views from two views by using the disparity-based view morphing technique, and the technique is very useful for multiple perspective video display systems.

2.2. Morphing Techniques

The morphing technique contains two transition functions for each corresponding pair in the source object and the target object. One is the forward transition function, which defines how the corresponding point moves from the source object to the target object. The other is the backward transition function, which defines how the corresponding point moves from the target object back to the source object. The forward transition function generates an intermediate object originating from the source object, and the backward transition function generates an intermediate object originating from the target object. In both functions, there are a variable s varying from 0 to 1, which determines how far the intermediate object is away from the source object. When s is 0, the position of the intermediate object is the same as that of the source object. When s is equal to 1, the position of the intermediate object is the same as that of the target object. The corresponding pairs of the forward transition function need not to be the same as those of the backward transition function. The morphing technique also contains a blending function, which determines how the intermediate objects of the forward and backward transition functions are merged into resulted transition objects.

Image morphing [6] is a special case of morphing, in which the object to be morphed is an image. The image morphing contains a set of functions which morph each pixel in the source image to a corresponding image point in the target image. However, the traditional image morphing function is usually a linear transition function of 2D image positions. If we simply apply the image morphing technique to an image-based virtual reality system, the generated transition views may not be the same as what the viewers should see if they walk from the source viewpoint to the destination viewpoint in the real world.

The view morphing [7] technique was originally proposed to generate realistic transition views for lateral transition function. View morphing contains the following three steps. The first step is prewarping. The fundamental matrix, which describes

the relation of the camera positions for taking the source and target images, can be estimated from the corresponding features in the source and target images. Then the source and target images are warped onto parallel image planes, where the warped image planes are both parallel to the transition baseline. The second step of view morphing is to apply the traditional image morphing (which uses linear transition function) to the warped images. After the transition images are generated by applying the linear image morphing, those images are then back-projected to the image planes, such that the resulted images are almost the same as the images taken by camera moving from the original source position and view direction to the original target position and view direction.

A new view morphing technique, called disparity-based view morphing, or abbreviated as "disparity morphing", has been proposed[8] to generate realistic transition views from two images for general view transition in image-based rendering systems. Disparity morphing renders the transition views based on the disparity map estimated from an image pair. In disparity estimation, epipolar constraints are used, not only for speeding up the searching of correspondences, but also for improving the accuracy of the estimated disparity maps. Then, the forward and backward morphing functions are defined based on the estimated disparity maps. Notice that the forward and backward morphing functions are usually not linear and not similar to the traditional image morphing functions. Finally, the interpolation function is defined to deal with the object occlusion problems. The resulted transition views look almost the same as what viewers should see in the real world.

3. MULTIPLE PERSPECTIVE VIDEO GENERATION

The details of the system for generating multiple perspective video are described in this section. Fig. 2 shows the block diagram of the video generation system, which can be divided into an encoder and a decoder. The original stereo videos captured by a set of stereo video cameras are the input data of our video generation system. The encoder prewarps the stereo video such that the video planes are parallel to each other, then estimates the correspondences of each pixel in the stereo video, generates the disparity maps, and then compresses the original stereo video as well as the generated disparity maps. These procedures of the encoder can be done in advance. The multiple views generator of the proposed system reads and decompresses the original videos and disparity maps, then applies the disparity morphing to generate desired views, and then performs the postwarping procedure to put the output video at the desired viewing direction. All the function block of the multiple perspective views generation system will be described in detail in the following subsections.

3.1. Prewarping

The purpose of the prewarping procedure is to make the image planes of original videos be parallel to each other and parallel to the stereo baseline between the optical centers of two cameras. As documented in previews research works, once the stereo images pair are parallel to each other and parallel to the stereo baseline, the correspondence estimation can be done by searching in the horizontal direction only, and the morphing function becomes linear. Searching in the horizontal direction simplifies the estimation procedure and increases the estimation accuracy. The linear morphing function simplifies the view generation procedure and increases the processing speed. Notice that, if the original stereo image planes are parallel (or almost parallel) to each other and to the baseline, the prewarping procedure can be omitted.

3.2. Object Segmentation

Because the computation disparity morphing is based on the estimated disparity maps, a reliable method for estimating accurate disparity is required for high quality image rendering. It is known that the texture can improve the matching accuracy and the object boundary may degrade the matching accuracy. However, it is not easy to distinguish the object boundary from the texture. To increase the matching accuracy we provide an interactive contour segmentation tool for extracting the object boundary information from the original stereo video, as illustrated in Fig. 3. We adopt the technique of active contours, or snake, in this interactive contour segmentation tool.

Snakes,[9–11] or active contours, are curves defined within an image domain that can move under the influence of internal forces coming from within the curve itself and external forces computed from the image data. The internal and external forces are combined so that the snake will deform to an object boundary or other desired features within an image. In Fig. 3(a), we specify some initial points for the contour model, and then obtain the segmentation result shown in Fig. 3(b). If there are some undesired portion appeared in the resulted contour, the interactive segmentation tool allows us to modify and correct them locally. Then, the resulted contour of the first frame can be tracked in the following video frames. Users are allowed to modify and correct the resulted contour at any video frame.

3.3. Disparity Estimation

Many techniques for disparity estimation, [12,13] or even for simultaneous occlusion detection, [14] from a stereoscopic image sequence have been proposed. In our system, we use adaptive-template matching, where the template size depends on local variation of image intensity and the knowledge of object boundary. Our method is divided into three steps: the adaptive-template matching, the disparity refinement, and the consistency check. The above three steps are performed for estimating both the left-to-right and right-to-left disparity maps. For simplicity, we shall describe only the estimation of left-to-right disparity.

3.3.1. Adaptive-template matching

A central problem in stereo matching by computing correlation or sum of squared differences lies in the selection of an appropriate template size. The template size must be large enough to include enough intensity variation for reliable matching but small enough to avoid the effects of projective distortion. For this reason, we propose a method for selecting the template size adaptively depending on local variations of image intensity and the knowledge of object boundary.

To cover enough intensity variation, we let the window size expand from the center pixel and set a threshold of gradient value for stopping the expansion of all paths along the vertical and horizontal directions from the central pixel. To avoid the effects of projective distortion, we also specify the largest template size allowed. We also use the object boundary information to prevent the template from covering different objects having large depth difference. Fig. 4 shows an example of the stereo image pair, which is generated by OpenGL for simulation and illustration purposes, and Fig. 5 shows the examples of the selected template in different image areas.

Since the image planes of the original stereo views we use in our experiments are almost parallel to each other, the corresponding point for each pixel can be searched horizontally based on the epipolar constraint. Since the search range is constrained along the horizontal direction and the template size is adaptively selected as described above, the accuracy of the corresponding point estimation can be improved a lot.

3.3.2. Disparity refinement

The result of disparity estimation using the adaptive-template matching may still be quite noisy. Therefore, a refinement algorithm is used to improve the matching result. The refinement algorithm is basically an iteration of two procedures, the median filtering and the global optimization. The median filter uses a cross-shape window size with 81 pixels to remove the outliers of the matching result. The use of the cross-shape window is to preserve the corner property. The global optimization minimizes an error function which is a weighted sum of the intensity difference for each corresponding pair and the disparity variation of adjacent pixels. The error function can be used to smooth the matching results. The function is defined as follows,

$$E = EI + r \times ED, \tag{1}$$

where *EI* is the sum of intensity difference of the correspondence defined in Eq. (2), *ED* is the disparity variation defined in Eq. (3), and *r* is the refinement factor specified by the user and is set as 1 in our experiments.

$$EI = \sum\nolimits_{x,y} \left\| I_L(p) - I_R(p') \right\|, \tag{2}$$

where $p = (x, y)$, $p' = C(x, y)$, $I(x, y)$ is the intensity of pixel at image position (x, y), $C(x, y)$ is the image position of the corresponding pixel (x, y), and $I_L(\cdot)$ and $I_L(\cdot)$ are the left and right images, respectively.

$$ED = \sum\nolimits_{p} \sum\nolimits_{q \in N(p)} (D(p) - D(q))^2, \tag{3}$$

where $D(x, y) = \left\| \overline{(x, y)C(x, y)} \right\|$ is the length of the disparity vector, or more precisely, the distance between corresponding points, and $N(x, y)$ is a set of the adjacent pixels of pixel (x, y).

Notice that the boundary information is also used in the above two refinement procedures to constrain the refinement is performed within the areas having similar depth values. The median filtering procedure and the smoothing optimization procedure are alternately performed until the resulted smoothness error value E is less than a threshold or the number of iteration reaches a preset value.

3.3.3. Consistency check of correspondence pairs

Occlusion is a difficult problem in computer vision, especially in stereo matching. Most matching methods tend to have erroneous results in the occluded area. Some researchers [14] use more than two images to reduce the correspondence errors. In this paper, we check the consistency of the correspondence pairs to detect the single occlusion area, as shown in Fig. 6(a), and then try to re-assign the correct disparity values. Single occlusion implies that the occluded area in the intermediate views only has information from one of the source stereo images. After adaptive-template matching and disparity refinement, we check if the left-to-right disparity for an image pair is consistent with the right-to-left disparity of its corresponding image point. If a point has no consistent corresponding point in the other image, we will regard this point as being in the occluded, and then assign to it the smaller disparity value of its adjacent points. The reason for such assignment is that the occluded point must be farther away than the object points that occlude it.

3.4. Data Compression

In our system, both the original stereo video and the disparity maps are compressed to reduce the storage size and the transfer time. Because there has already been many fast algorithms designed for the MPEG compression standard, we simply use the MPEG [15] standard to compress and decompress the video signals.

The disparity maps are compressed losslessly by using the compression technique similar to the lossless mode of the JPEG standard. The disparity of each corresponding pixel is firstly predicted by its three neighboring pixels, as Fig. 7, and then the prediction error is encoded by the Huffman coding technique

3.5. Novel Views Generation

In the rendering step, we use the disparity morphing technique to on-line generate multiple perspective stereo video from the two original views and the two disparity maps. After the stereo video and the disparity maps are decompressed, we first convert each disparity map to a layered depth structure. The layered depth structure is obtained by applying run-length encoding. Next, the image point to be generated can be computed by linear interpolation or extrapolation from the corresponding pixel positions with a viewpoint control parameter s. The control parameter s defines the location of the viewpoint. When s is 0, the viewpoint is located at the original left-eye viewpoint, and when s is 1, the viewpoint is located at the original right-eye viewpoint. Fig. 8 summaries the meaning of the control parameter s. By organizing the disparity map into a layered depth structure, we can save some overhead in rendering. Using the depth ordering of the layered depth structure, we do not need the Z-buffer (disparity-buffer) any more. Because a layer in this structure has the same depth, we can calculate the warping offset once for the whole layer and move the data to the right position layer by layer, instead of pixel by pixel. In general, the number of pixels is at least several times larger than the number of runs in an image. Furthermore, we use multi-thread programming to improve the performance of the whole system.

The generated video usually contains holes and occluded objects. In our system, the holes can be eliminated by interpolating the values of its adjacent pixels, and the object occlusion can be dealt with correctly by analyzing the disparity flows as described below. Fig. 9 shows an example of disparity flows. In this example, there is a background object ranging from points *p1* to *p6* occluded by a foreground object located at points *p3* and *p4*. Thus, the points *p3* and *p4* moves faster than *p1*, *p2*, *p5*, and *p6*. Consider $s = 0.5$ with the view interpolation procedure, point *p3* and *p5* will morph to the same image position at $s = 0.5$. In the traditional image morphing, the pixel value generated is usually a weighted sum of the values of the original points *p3* and *p5*, which will generate ghost effects in the resulted image. With our method, the disparity of each corresponding point is examined to decide which point value should be used as the resulted pixel value. In this example, the disparity of point *p3* is larger then that of point *p5*, which means the object at point *p3* is closer to the camera than the object at the *p5*. Hence, the pixel value of point *p3*, the point with larger disparity, is used as the resulted pixel value. By applying this algorithm, the view generation method can deal with the object occlusion problem correctly, and can generate better novel views..

3.6. Postwarping

The postwarping procedure warps the above generated novel view to the desired viewing direction. The desired viewing direction is usually pointing toward to the interested area in the scene. Usually, no postwarping procedure is required if the prewarping can be omitted.

4. EXPERIMENTAL RESULTS

This section shows the experimental results of applying the proposed algorithm to the synthetic and real stereo videos. When applying to the synthetic stereo video, we can see the differences between the generated views and the true views which one is supposed to see. Fig. 10 shows such an example, where the result images are scaled up to show the details. In Fig. 10, we can see that the generated view, either using the true disparity information or the estimated disparity map are almost the same as the true view. Fig. 11 shows the experimental results using the real stereo video data. The second row of Fig. 11 shows the generated views of the high-ambiguity area, and the third row shows the generated views of the occluded area. In the high-ambiguity case, we can see there are some noises in the generated views caused by the inaccuracy of the estimated disparity information. It also shows the double occlusion area, which is explained in Fig. 6(b). For double occlusion area, no information is available for rendering the occluded area in the intermediate views, and we can only use the information of the adjacent pixels to generate the intensity value. From Fig. 11(g)-(j), we can see that our system can handle single occlusion correctly. From the above experimental results, we can see that the areas near the boundary may not perform very well. This is because we use the pixel-wise disparity information, and the quantization during the interpolation and extrapolation will affect the rendering quality. Alpha blending may be a good solution to this problem, which will be our future work.

5. CONCLUSION AND FUTURE WORK

A system for generating multiple perspective stereo video from a two-view video sequence are presented. Our system first estimates the corresponding point of each pixel by adaptive-template matching, and then constructs two disparity maps for each image pair. Next, each disparity map is converted into a layered depth structure. Based on the layered depth structures, the system can generate multiple perspective stereo video by interpolation or extrapolation from the original views. Both the disparity maps and the original stereo videos are compressed to reduce the storage size and the transfer time. With the layered depth structures and multi-thread programming, our system can generate, in real time, a stereo video of multiple perspective view in response to the head movement of the user. Compared to the traditional method of capturing multiple perspective video directly, the approach of view interpolation can eliminate the problems caused by the requirement of synchronizing multi-channel video input and by the large amount of video data needed to be stored and transferred.

In the future, the stereo video may be coded by a block-based stereoscopic coding scheme using disparity, which has been proposed as an MPEG standard for the coding of stereoscopic image sequences. The interframe disparity coding will also be addressed in the future. Furthermore, the relation between the resolution of disparity maps and the performance of the generated multiple views is another interesting problem. To reduce the amount of required data, we can decrease the run number of the layered depth structure until the decrease of the disparity resolution will result in unacceptable image quality in the generated views. Another problem is how to improve the pixel-wise disparity morphing as mentioned in Section 4.

ACKNOWLEDGMENTS

This work is partially supported by Opto-Electronics & Systems Laboratory, Industrial Technology Research Institute, Hsinchu, Taiwan, ROC.

REFERENCES

1. Et al. S. Moezzi, "Immersive video," In *Proceedings of the IEEE Virtual Reality Annual International Symposium*, March 1996.
2. Et al. S. Moezzi, "Reality modeling and visualization from multiple video sequences," *IEEE Computer Graphics and Applications*, pp. 58-63, November 1996.
3. S. E. Chen, "QuickTime VR — an image-based approach to virtual environment navigation," *SIGGRAPH '95*, pp. 29-38, 1995.
4. H.-C. Huang and Y.-P. Hung, "Spisy: The stereo panoramic imaging system," In *RAMS'97, Third Workshop on Real-Time and Media Systems*, pp. 71-78, Taipei, Taiwan, R.O.C., July 1997.

5. H.-C. Huang, C.-C. Kao, and Y.-P. Hung, "Generation of Multiviewpoint Video from Stereoscopic Video," *IEEE Trans. Consumer Electronics*, vol. 45, pp. 124-133, 1999.
6. T. Beiter and S. Neely, "Feature-based image metamorphosis," *Proceedings SIGGRAPH '92*, pp. 35-42, 1992.
7. S. M. Seitz and C. R. Dyer, "View morphing," *Proceedings SIGGRAPH '96*, pp. 21-30, 1996.
8. H.-C. Huang, S.-H. Nain, Y.-P Hung, and T. Cheng, "Disparity-based — a new technique for image-based rendering," In ACM symposium on Virtual Reality Software and Technology 1998(VRST'98), pp. 9-16, Taipei, Taiwan, R.O.C., November 1998.
9. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision*, vol. 1, pp. 321-331, 1987.
10. W. M. Neuenschwander, P. Fua, L. Iverson, G. Szekely, and O. Kubler, "Ziplock Snakes," I*nternational Journal of Computer Vision*, vol. 25, pp. 191-201, 1997.
11. C. Xu, and J. L. Prince, "Snakes, Shapes, and Gradient Vector Flow," *IEEE Trans. Image Processing*, vol. 7, pp. 359-369, 1998.
12. Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 7, pp. 139-154, 1985.
13. C.-J. Tsai and A. K. Katsaggelos, "Dense Disparity Estimation with a Divide-and-Conquer Disparity Space Image Technique," *IEEE Trans. Multimedia*, vol. 1, pp. 18-19, 1999.
14. N. Grammalidis and M. G. Strintzis, "Disparity and Occlusion Estimation in Multiocular Systems and Their Coding for the Communication of Multiview Image Sequences," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, pp. 328-344, 1998.
15. *MPEG standard, ISO-IEC/JTC1 SC29 on 22*, November 1991.

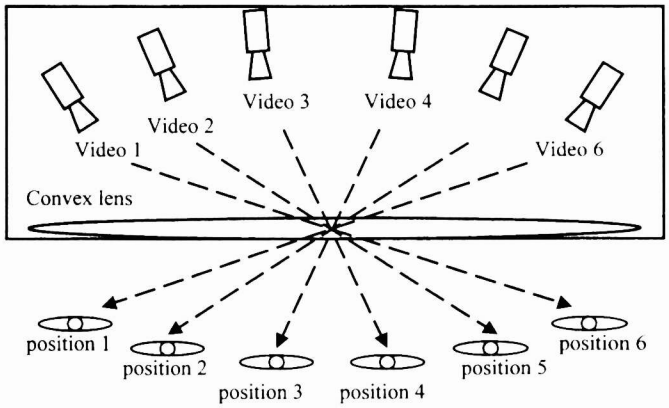


Figure 1. The multiple perspective video display system.

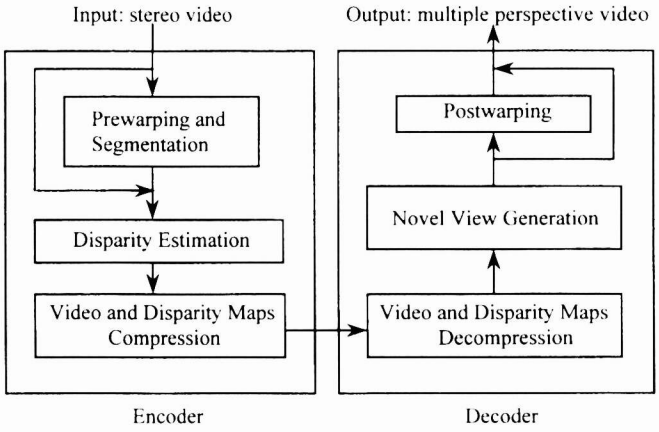


Figure 2. The block diagram of the multiple perspective video generation system.

Figure 3. An example of object segmentation: (a) The given initial contour, (b) The segmentation result, (c) Tracking after fifteen frames.

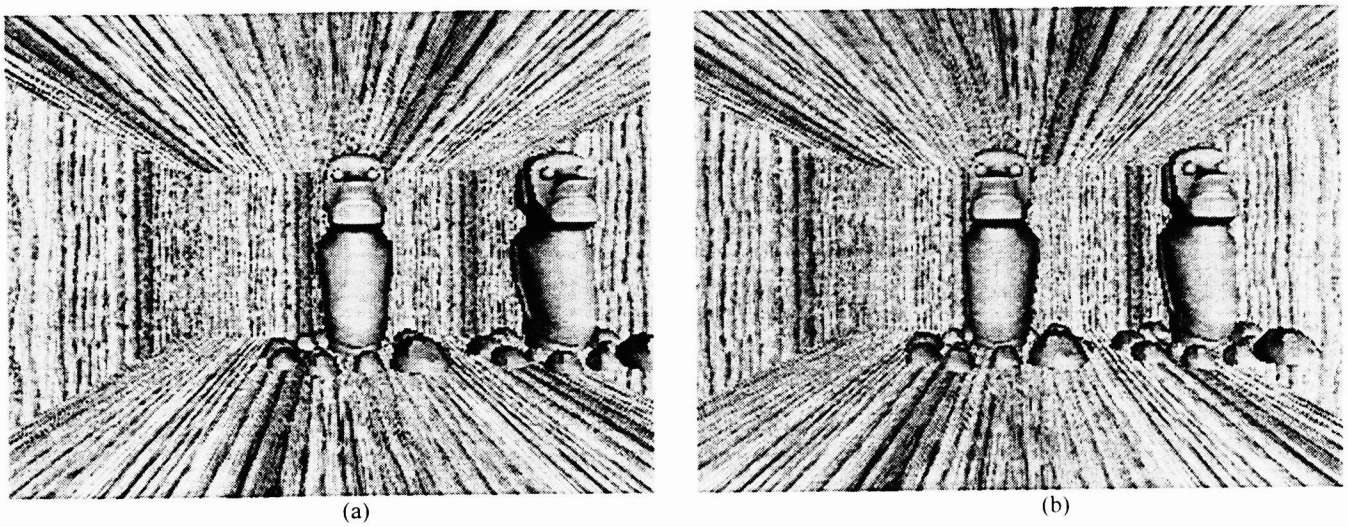


Figure 4. An example of the synthetic stereo image pair: (a) the left image, (b) the right image.

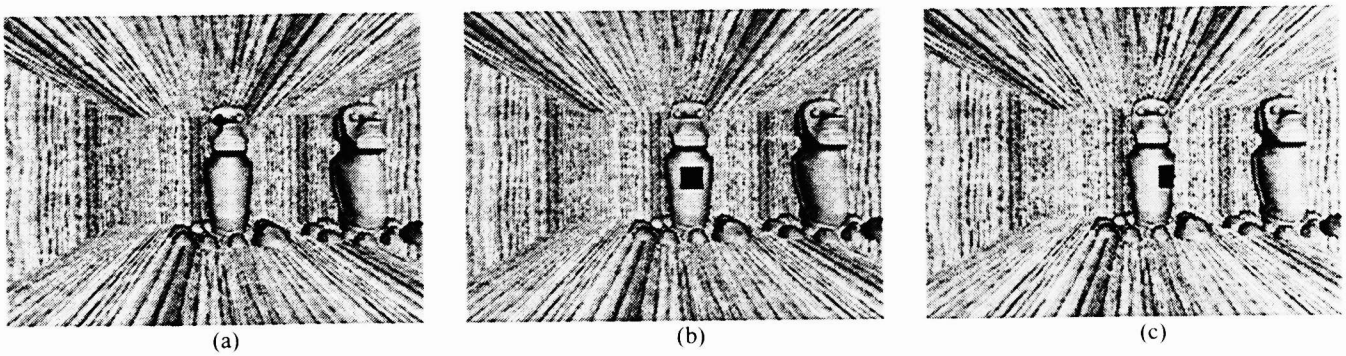


Figure 5. Some examples of the selected templates in the different image areas: (a) the area with large intensity variation, (b) the homogeneous area, (c) the homogeneous area under boundary constraint.

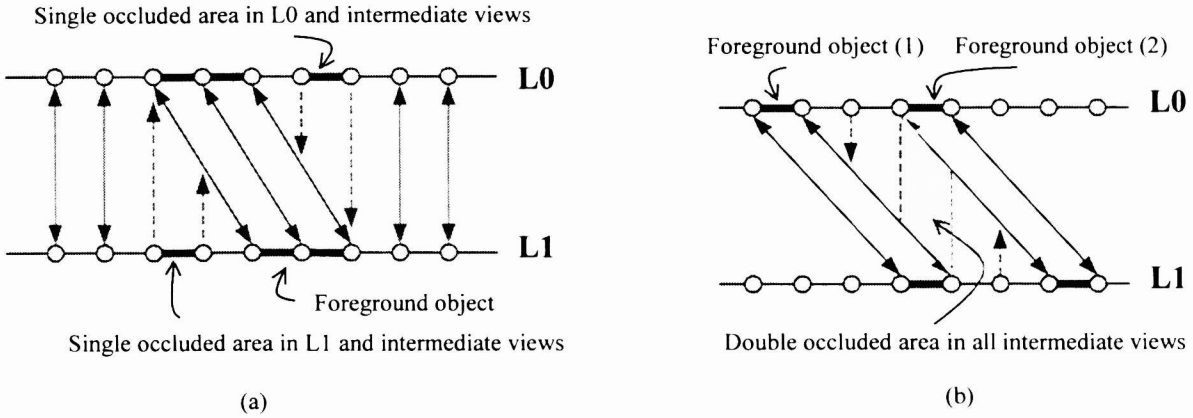
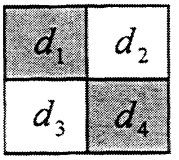


Figure 6. Two cases of object occlusion, L0 and L1 are epipolar line pair: (a) Single occlusion, (b) double occlusion.



$$d_1 + d_3 \approx d_2 + d_4$$

Figure 7. Disparity prediction: The predicted value of d_4 :

$$d_4^* = d_2 + d_3 - d_1 .$$

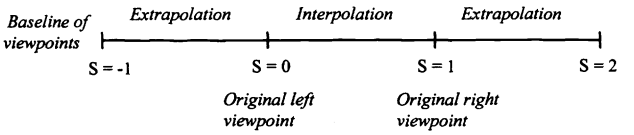


Figure 8. The meaning of the viewpoint control parameter *s*.

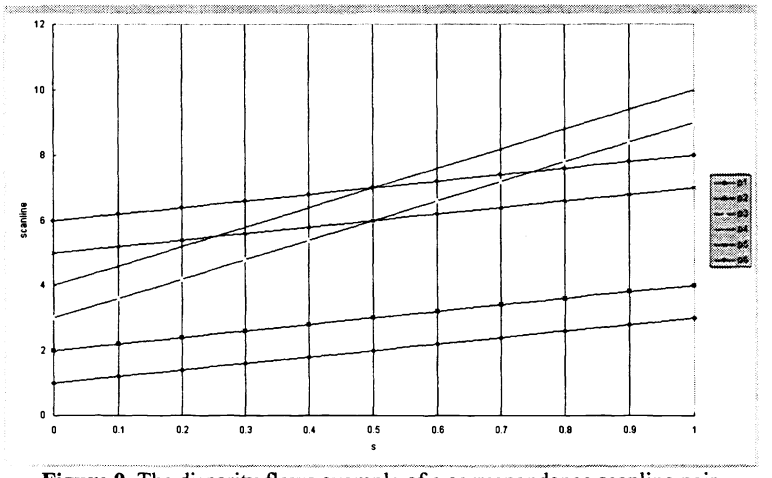
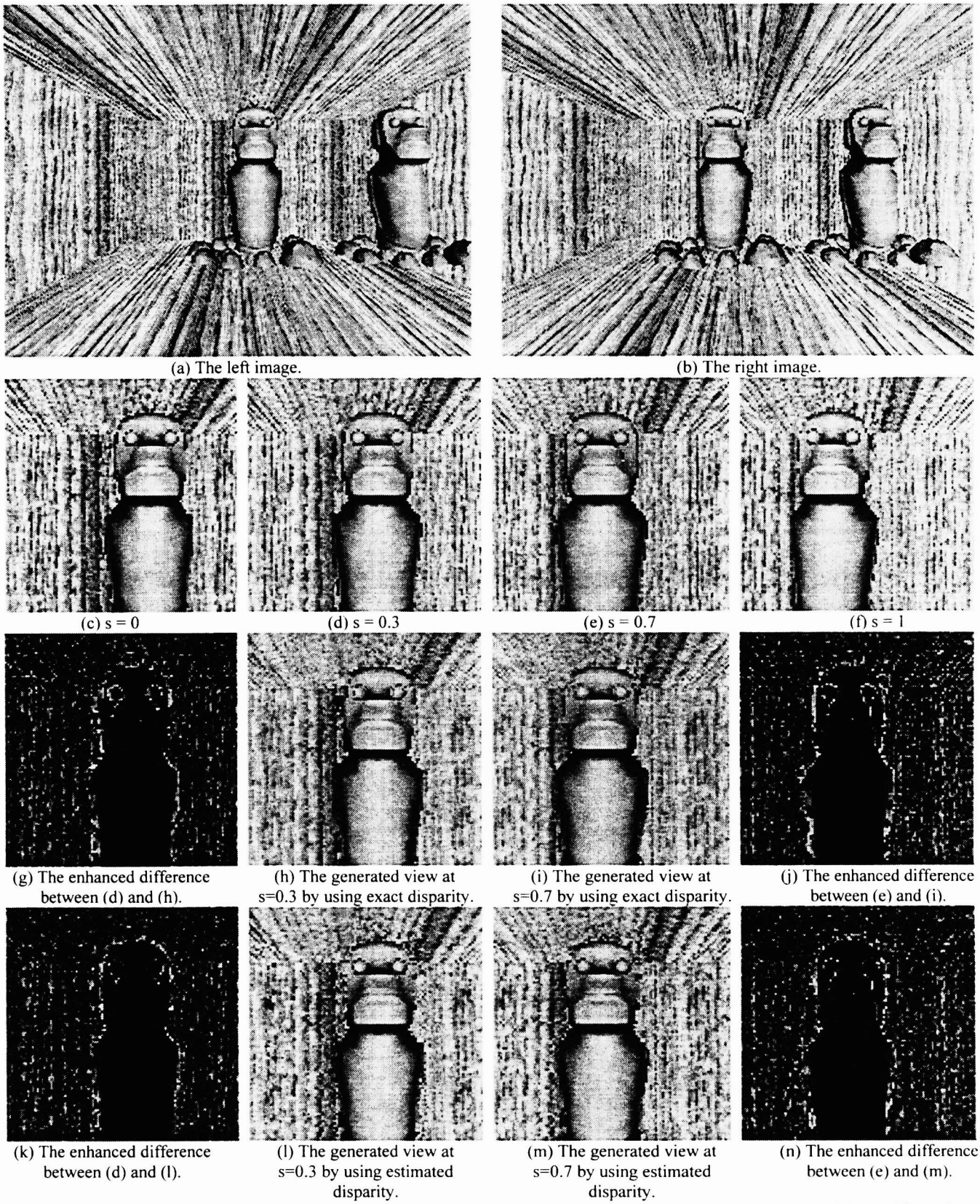


Figure 9. The disparity flows example of a correspondence scanline pair.



(a) The left image. (b) The right image.

(c) s = 0 (d) s = 0.3 (e) s = 0.7 (f) s = 1

(g) The enhanced difference between (d) and (h). (h) The generated view at s=0.3 by using exact disparity. (i) The generated view at s=0.7 by using exact disparity. (j) The enhanced difference between (e) and (i).

(k) The enhanced difference between (d) and (l). (l) The generated view at s=0.3 by using estimated disparity. (m) The generated view at s=0.7 by using estimated disparity. (n) The enhanced difference between (e) and (m).

Figure 10. Experimental results using synthetic stereo video: (a), (b) The stereo image pair, (c), (d), (e), (f) The partial magnifiers of true views, (h), (i), (l), (m) The generated views, (g),(j),(k),(n) The enhanced difference between the true views and the generated views.

Figure 11. Experimental results of the real scene stereo video, (a),(b) The stereo image pair, (c),(f) The partial magnifiers of the high-ambiguity area in the source stereo image pair, (d), (e) The generated views of the high-ambiguity area, (g), (j) The partial magnifiers of the occluded area in the source stereo image pair, (h),(i) The generated views of the occluded area.