Short Paper

Real-Time Software Method for Preserving Eye Contact in Video Conferencing

YU-PAO TSAI[1,3], CHING-CHE KAO[1,2], YI-PING HUNG[1,2] AND ZEN-CHUNG SHIH[3]
[1]*Institute of Information Science*
Academia Sinica
Taipei, 115 Taiwan
[2]*Department of Computer Science and Information Engineering*
National Taiwan University
Taipei, 106 Taiwan
E-mail: hung@csie.ntu.edu.tw
[3]*Department of Computer and Information Science*
National Chiao Tung University
Hsinchu, 300 Taiwan

By maintaining eye contact in visual communication, such as in video conferencing, the feeling of intimacy can be greatly enhanced. Unfortunately, due to the physical constraints on the arrangement of cameras and monitors, it is not easy to preserve eye contact in video conferencing. Most people have tried to solve this problem using hardware approaches, which usually lead to bulky or expensive solutions. In this paper, we propose a real-time software method for preserving eye contact in video conferencing. For each station involved in video conferencing, we use two pre-calibrated cameras to acquire a stereo view of the local participant. Then, we apply disparity-based view morphing to the acquired stereo view to generate the desired view that preserves eye contact with another participant. Our experiments demonstrate that the eye contact problem can indeed be solved *in real-time* by using our software approach. This is the first time a software approach has been shown to be practical for preserving eye contact in video conferencing.

Keywords: video conferencing, eye contact, disparity estimation, disparity-based view morphing, eye tracking

1. INTRODUCTION

In visual communication, such as in video conferencing and in videophone applications, the feeling of intimacy between participants at different stations can be greatly enhanced by maintaining eye contact between them during a conversation. However, it is

Received April 19, 2002; revised November 11, 2002 & February 6, 2003; accepted March 18, 2003.
Communicated by Kuo-Chin Fan.

not easy to preserve eye contact in video conferencing due to physical constraints on the arrangement of cameras and monitors, as explained below.

In a video conferencing system, the video camera and the display monitor are two major components that should function together. The camera acquires a sequence of head-on images of the speaker (i.e., the local participant) and then transmits the images to the listener (i.e., the remote participant) through the communication network. At the same time, the speaker may be watching the image of the listener that is displayed on the monitor. However, due to the arrangement of the camera and monitor, the listener may feel that the speaker is not looking at him, even though the speaker is indeed looking at the image of the listener. Fig. 1 illustrates the eye contact problem described above.

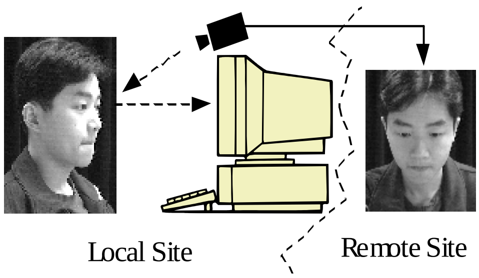


Fig. 1. Illustration of the eye contact problem.

To preserve eye contact, it may be desired that the camera be placed at the position where the images of the listener are displayed. However, the camera and the monitor cannot both be placed at the same location physically. One popular solution is to place a half-transparent mirror in front of the speaker, as illustrated in Fig. 2. The half-transparent mirror is oriented at 45 degrees to the gaze direction of the speaker, and the camera and monitor are placed orthogonally so that the image of the monitor coincides with the camera [10]. Many other hardware methods using half-transparent mirror-like apparatuses have also been proposed [8, 11, 13, 20]. However, the display devices involved in hardware approaches are usually bulky, awkward, and/or expensive. Another disadvantage is that the half-transparent mirror attenuates both the light emitted from the monitor and the light transmitted to the camera, hence causing the quality of the perceived images to deteriorate.

To avoid the disadvantages of hardware approaches, some researchers have tried software approaches [12, 15]. Unfortunately, their methods are not suitable for real-time rendering and, hence, are not practical for preserving eye contact in video conferencing. In this paper, we propose a real-time software method for preserving eye contact in video conferencing. A prototype system has been successfully implemented to demonstrate the real-time performance of our software method. Fig. 3 shows the block diagram of the prototype system. For each station involved in video conferencing, we use two pre-calibrated cameras to acquire a stereo view of the local participant. The acquired stereo view can be easily warped (or rectified) by using pre-calibrated camera parameters. Then, we utilize disparity-based view morphing [5] to generate the desired view that pre-

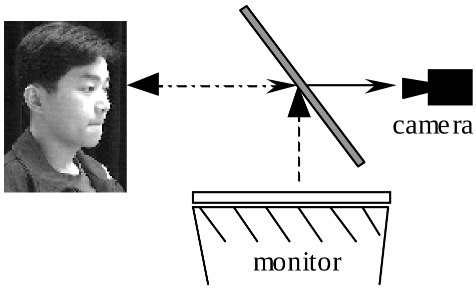


Fig. 2. Illustration of a hardware approach.

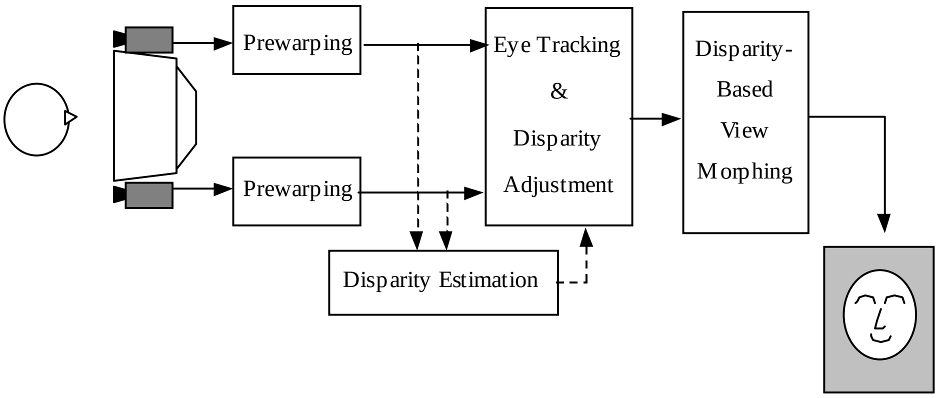


Fig. 3. Block diagram of our system for reconstructing eye contact in video conferencing.

serves eye contact. To apply disparity-based view morphing, we need to first estimate left-to-right and right-to-left disparity maps. Because disparity estimation is a time consuming process, we only estimate the disparity maps for the first stereo view, i.e., only at the initialization stage. Once the disparity maps are obtained, the process of generating the desired view from the acquired stereo view consists of four steps: prewarping, eye tracking, adjustment of disparity maps, and novel view generation. We will describe each step of our system, and also the method for disparity map estimation, in the following sections.

The remainder of this paper is organized as follows. The technique we use to estimate disparity maps is described in section 2. Section 3 describes in detail the proposed method for preserving eye contact, which includes prewarping, eye tracking, adjustment of disparity maps, and novel view generation using disparity-based view morphing. Experimental results are presented in section 4, and conclusions and future works are given in section 5.

2. DISPARITY ESTIMATION

May methods have been proposed for estimating disparity between two images. Two widely used criteria for matching are photometric similarity and spatial smoothness. There are many existing techniques for disparity estimation, including block matching [1, 2, 6], regularization techniques [16, 21], intrinsic curve transformation [22], and dynamic programming based techniques [14, 23, 24]. However, the problem of object occlusion is hard to deal with. Some multiple camera algorithms [3, 4, 17, 19] have been presented for disparity estimation and simultaneous occlusion detection. Extra cameras are employed to provide more information to enrich the matching cost function. While these techniques reduce the occurrence of occlusion and improve disparity estimation performance through the use of multiple views of the scene, they increase the camera complexity and the bandwidth required to transmit additional views.

In this section, we describe a new method for disparity estimation, which can make use of the knowledge of object boundary, if available. Our method is divided into three steps: template matching with adaptive windows, disparity refinement, and consistency check. The block diagram is shown in Fig. 4. The above three steps are performed twice to estimate both the left-to-right and right-to-left disparity maps. For simplicity, we shall describe only the process of estimating left-to-right disparity.

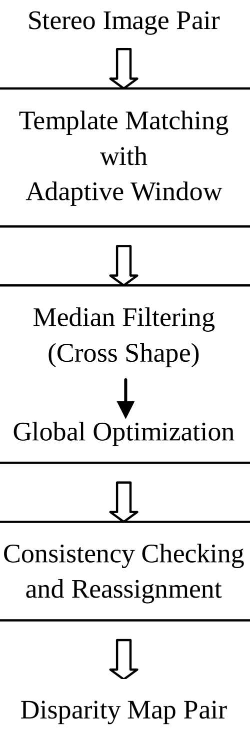


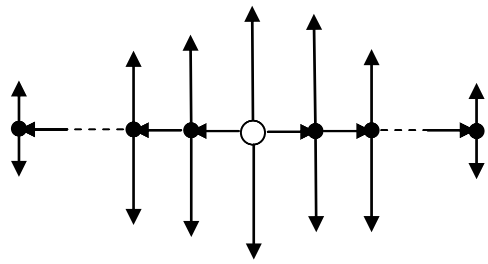
Fig. 4. The block diagram of disparity estimation.

2.1 Template Matching with Adaptive Windows

In the case of stereo matching methods performed using correlation or the sum of squared differences, a critical issue is 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, Kandade and Okutomi proposed a stereo matching algorithm with an adaptive window in 1994 [6]. They presented a method to select an appropriate window by evaluating the local variation of the intensity and disparity. They employed a statistical model of the disparity distribution within the window and then iteratively updated the disparity estimation for each point by choosing the size and shape of the window with the least uncertainty until it converged. In this work, we propose a fast algorithm for selecting the template size adaptively, depending on local variation of the image intensity.

To cover enough intensity variation, we let the window size expand from the center pixel along a set of pre-selected paths. Then, we set a threshold for the accumulative gradient value to stop the expansion of the paths along the vertical and horizontal directions from the central pixel. Based on our experience, the threshold of the accumulative gradient should be set to 192 for color images (RGB) and 64 for gray images. To avoid the effects of projective distortion, we also specify the largest template size allowed, e.g., 7×7 or 9×9. Fig. 5 (a) shows an example of pre-selected paths for window expansion, which first travels horizontally and then vertically. Fig. 5 (b) shows an example consisting of the intensities of the pixels included in an allowed maximum window centered on image pixel X, and Fig. 5 (c) shows the accumulative gradient values and the expanded template window, enclosed by the red lines, for pixel X. At the beginning, the accumulative gradient value of X is set to 0. Following the pre-selected path shown in Fig. 5 (a), we start to expand upward. The first intensity value we meet is 148, so the accumulative gradient value is $0 = 0 + |148 - 148|$. The accumulative gradient value is smaller than the threshold of the accumulative gradient, e.g., 64 in this example; hence, the expansion continues. The next one we meet is 99, and its accumulative value is $13 = 0 + |135 - 148|$. It still is not larger than 64. Next, we meet 83 and calculate its accumulative gradient value. Its accumulative gradient value is $65 = 13 + |83 - 135|$ and is indeed larger than 64. Therefore, we do not include it in the expanded window and stop the expansion along this path. Next, the adaptive window is expanded downward, i.e., $148 \rightarrow 146 \rightarrow 134 \rightarrow 132$. The corresponding accumulative gradient values are shown in Fig. 5 (c). Because none of them are larger than 64, the expansion does not stop until it meets the maximum size limit. Following the above procedure, we can obtain the expanded window shown in Fig. 5 (c). It should be noted that we also use object boundary information, if it is available, to prevent the template from covering different objects having large depth differences.

Fig. 6 shows a stereo image pair of a real scene. The stereo matching results obtained by using the simple block matching method are shown in Figs. 7 (a) and 7 (b), and Figs. 7 (c) and 7 (d) shows the stereo matching results obtained by using our method of template matching with an adaptive window. The allowed largest window size was set to be 9×9 in this example. Also, we let 3×3 be the minimal window size to avoid ambiguous matching when the texture in the template was barely uniform. Figs. 7 (a) and 7 (c) are the left-to-right disparity maps, which record the horizontal disparity between the



(a)

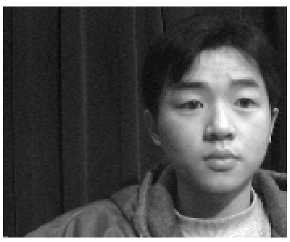
162	171	162	83	91	136	147
155	159	159	135	70	72	106
145	143	149	148	119	59	81
134	139	146	X 148	160	147	154
133	133	136	146	160	172	172
132	129	132	134	151	161	166
125	130	128	132	136	146	147

(b)

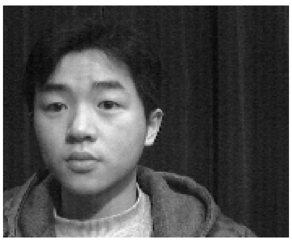
42	41	18	-	-	-	-
35	29	15	13	-	-	-
25	13	5	0	53	-	-
14	9	2	0	12	25	32
15	15	12	2	12	50	50
16	19	16	14	21	61	56
23	20	20	16	36	-	-

(c)

Fig. 5. An example illustrating expansion of the adaptive window. The arrows shown in (a) indicate the direction for accumulating the gradient values. (b) shows the intensities of the pixels included in the maximum window, e.g., 7×7, centered on image pixel X. (c) shows the accumulative gradient value and the expanded window for pixel X shown in (b). In this example, we set the threshold of the accumulative gradient value to be 64.



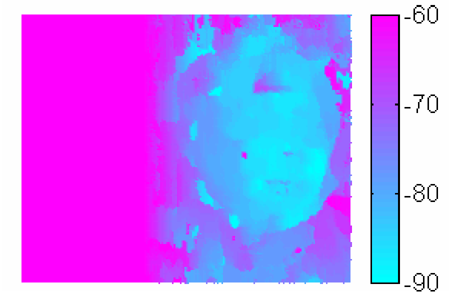
(a)



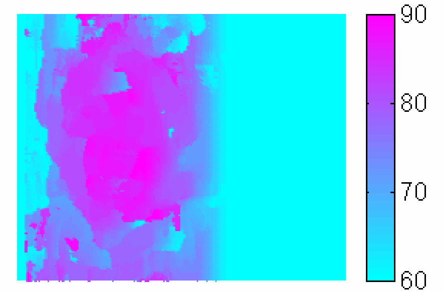
(b)

Fig. 6. A stereo image pair. (a) The left image; (b) The right image.

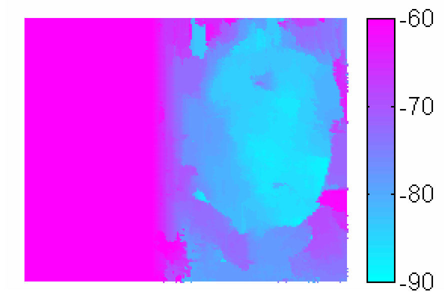
left image and the right image. Similarly, Figs. 7 (b) and 7 (d) are the right-to-left disparity maps, which record the horizontal disparity between the right image and the left image. In this paper, we use different colors to represent different disparity values. The color bar shown on the right side of each disparity map depicts the corresponding disparity value of the disparity map. We can easily notice that there are obvious differences between these two results. Compared with the simple block matching method, our method greatly improves the accuracy of the corresponding point estimation.



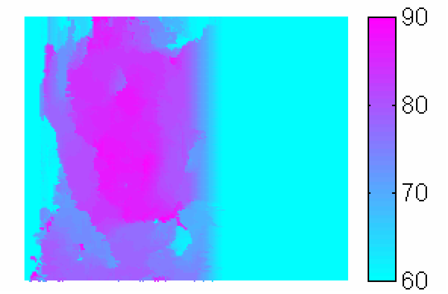
(a) Left-to-right.



(b) Right-to-left.



(c) Left-to-right.



(d) Right-to-left.

Fig. 7. (a) and (b) show a pair of disparity maps generated by using the simple block matching method. (c) and (d) show a pair of disparity maps generated by using our method of template matching with an adaptive window.

2.2 Disparity Refinement

The result of disparity estimation obtained using template matching may occasionally be very noisy. Therefore, to improve the quality of the disparity estimates, a refinement algorithm should be used. Our refinement algorithm includes two procedures, median filtering and global optimization. Median filtering uses a cross-shape window having a size of 81 pixels, as shown in Fig. 8, to remove the outliers of the matching result. The purpose of using the cross-shaped window is to preserve the corner property. Fig. 9 shows the disparity maps obtained after applying median filtering to the estimated disparity maps shown in Figs. 7 (c) and 7 (d).

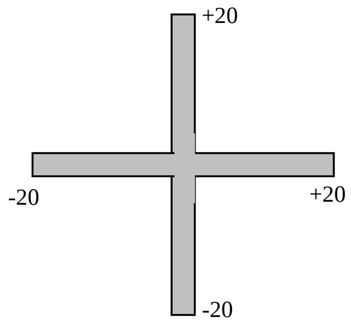
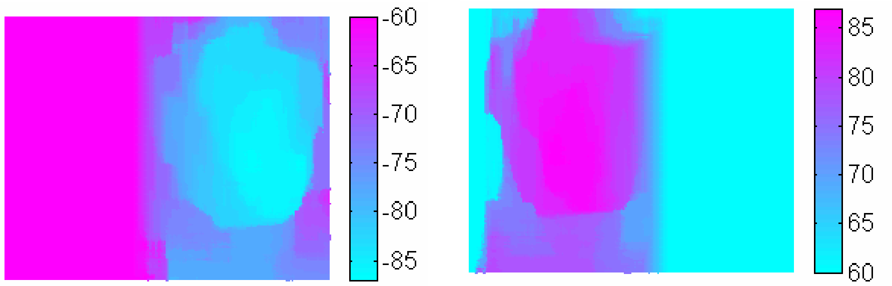


Fig. 8. The cross-shaped window consisting of 81 pixels for median filtering.



(a) Left-to-right. (b) Right-to-left.

Fig. 9. The disparity maps after applying median filtering to the disparity maps shown in Figs. 7 (c) and 7 (d).

The subsequent global optimization procedure is used to further refine the matching results. To simplify our explanation, we refer to the pixels in an $M \times N$ image with a single index k, where k takes on values in $\{1, 2, \ldots, M \times N\}$. Let $\mathbf{d} = (d_1, d_2, \ldots, d_{M \times N})^t$ be the disparity vector composed of all the disparity values. The goal of the global optimization process is to find the optimal disparity vector $\mathbf{d}^*$ by minimizing the following error function $J(\mathbf{d})$:

$$J(\mathbf{d}) = \alpha EI(\mathbf{d}) + (1 - \alpha) ED(\mathbf{d}), \tag{1}$$

where EI is the sum of the squared intensity difference of corresponding points, as defined in Eq. (2), and ED is the sum of the squared disparity variation of adjacent pixels, as defined in Eq. (3). Here, α is a factor specified by the user, and it was set as 0.05 in our experiments. The error term due to the intensity difference is defined as

$$EI(\mathbf{d}) = \sum_{k=1}^{M \times N} \left\| I_L(\mathbf{p}_k) - I_R(\mathbf{p}_k + \begin{bmatrix} d_k \\ 0 \end{bmatrix}) \right\|^2, \tag{2}$$

where $I_L(\bullet)$ and $I_R(\bullet)$ are the left and right images, respectively. Here, the y-component of the disparity is zero because we have prewarped the images to correct the perspective distortion, as explained in section 3.1. The error term due to disparity variation is defined as

$$ED(\mathbf{d}) = \sum_{k=1}^{M \times N} \sum_{\mathbf{p}_j \in N(\mathbf{p}_k)} (d_k - d_j)^2 \cdot \delta(d_k, d_j), \tag{3}$$

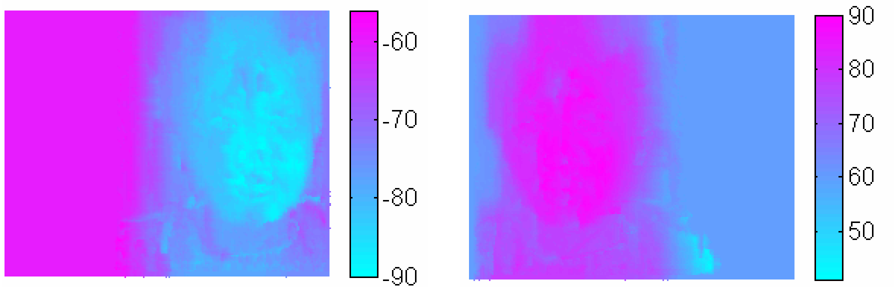
where $N(\mathbf{p}_k)$ is a set of adjacent pixels (e.g., four- neighborhood) of pixel $\mathbf{p}_k$. Here, $\delta(d_k, d_j)$ implements the commonly used line process of the discontinuity constraint: When the disparity difference between pixel $\mathbf{p}_k$ and pixel $\mathbf{p}_j$ is larger than a threshold, the value of $\delta(d_k, d_j)$ will be set to zero. Otherwise, it will be set to one. In this work, we choose the threshold for $\delta(d_k, d_j)$ to be a quarter of the width, i.e., M/4, based on our experience.

To solve for the optimal solution $\mathbf{d}^*$, a brute force algorithm which exhaustively examines all possible realizations of **d** can be applied. However, this is not a practical approach due to the formidable computational cost. Therefore, the optimal solution is usually computed by using some iterative search techniques. In this study, we adopted the iterated conditional modes (ICM) method, which converges very fast [9]. In each iteration, the disparity value of a selected pixel $\mathbf{p}_k$ is updated by minimizing the following equation:

$$d_k = \arg \min_{d_{\min} \le d \le d_{\max}} (\alpha EI'(d;\mathbf{p}_k) + (1-\alpha) ED'(d;\mathbf{p}_k)), \tag{4}$$

where $EI'(d;\mathbf{p}_k) = \left\| I_L(\mathbf{p}_k) - I_R(\mathbf{p}_k + \begin{bmatrix} d \\ 0 \end{bmatrix}) \right\|^2$, $ED'(d;\mathbf{p}_k) = \sum_{\mathbf{p}_j \in N(\mathbf{p}_k)} (d - d_j)^2 \cdot \delta(d, d_j)$, d_{min} is the minimum disparity value, and d_{max} is the maximum disparity value.

Fig. 10 shows the resulted disparity maps obtained after applying the global optimization procedure to the disparity maps shown in Fig. 9.



(a) Left-to-right. (b) Right-to-left.

Fig. 10. The disparity maps obtained after applying global optimization to the maps shown in Fig. 9.

2.3 Consistency Checking of Correspondence Pairs

Occlusion is a difficult problem in computer vision, especially in stereo matching. Most matching methods tend to produce erroneous results in the occluded area. In our method, we check the consistency of the correspondence pairs to detect the single occlusion area, as illustrated in Fig. 11, and then try to re-assign the correct disparity values through extrapolation using neighboring background pixels. A simple implementation is described as follows. Single occlusion implies that the occluded area in the intermediate views only has information from one of the two source images. After template matching with an adaptive window and disparity refinement are performed, 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. For each pixel in both images, we add the disparity vector in one direction to the corresponding disparity vector in the opposite direction (ideally, this sum should be zero). If the absolute value of the sum is greater than a threshold, we will regard this pixel as an occluded pixel and simply assign to it the smaller disparity value of its neighboring unoccluded points. The reason for such assignment is that the occluded

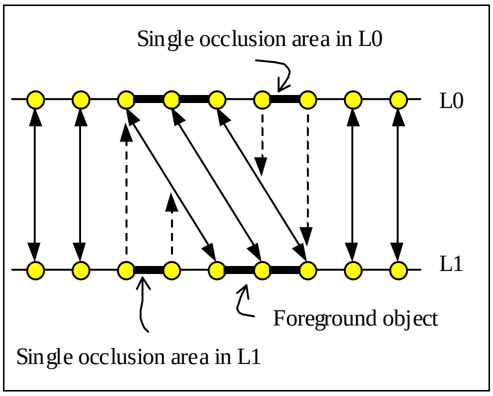
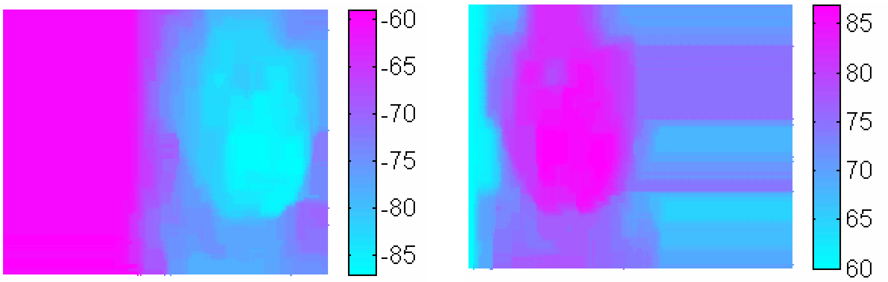


Fig. 11. Illustration of the single occlusion case. Here, L0 and L1 represent a pair of epipolar lines.



(a) Left-to-right. (b) Right-to-left.

Fig. 12. The disparity maps obtained after applying consistency checking to the maps shown in Fig. 10.

point must be farther away than the object point that occludes it. The threshold can be derived from statistics of disparity estimates for occluded and unoccluded regions. Fig. 12 shows the disparity map pair obtained after applying the consistency check procedure to the disparity maps shown in Fig. 10.

3. RECONSTRUCTION OF EYE CONTACT

This section describes in detail the proposed method for reconstructing eye contact in video conferencing. The block diagram of our prototype system has been shown in Fig. 3. After the disparity maps are estimated by using the first stereo view at the initialization stage, the proposed method for preserving eye contact can be performed in four steps: prewarping, eye tracking, adjustment of disparity maps, and novel view generation.

3.1 Prewarping

In our current implementation, we first calibrate the two cameras mounted on both sides of the monitor [18]. The known object used for camera calibration is shown in

Fig. 13. The calibration procedure is briefly described in the following. First, we fix the position of the stereo cameras and capture a set of images of the calibration plate while the calibration plate is moving along a translation axis with known movements. Second, we extract the centroids of the white circles on the calibrate plate and use them to estimate the camera parameters. Once the camera parameters are known, the stereo images can be warped, and they form the so-call "rectified stereo view." The image planes of the rectified stereo view should be parallel to each other and parallel to the stereo baseline. Therefore, we can apply the linear morphing function to generate the desired view that preserves eye contact.

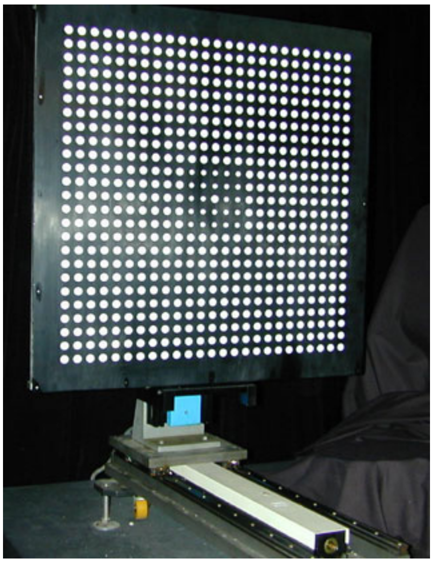


Fig. 13. The calibration object used in this work.

3.2 Eye Tracking and Disparity Adjustment

As we mentioned above, disparity estimation is a time-consuming process. For a real-time system, it is almost impossible to estimate the disparity maps for each frame. Instead, we use pattern tracking and disparity map adjustment. The idea is that the disparity maps can be considered as a kind of 3D description of the speaker's face. If we can obtain the speaker's motion information by tracking some special pattern on the speaker's face, then we can use this motion information to adjust (or "relocate") the reference disparity maps accordingly.

In our current implementation, we simply use the speaker's eyes as the pattern for tracking, as illustrated in Fig. 14. The simple three-step algorithm is used to estimate the motion vector of the speaker. Then, we adjust the reference disparity maps, which are estimated at the initialization stage, according to the estimated motion information of the speaker. The adjusted disparity maps can then be used to generate the desired view that preserves eye contact in video conferencing.

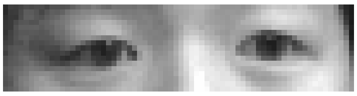


Fig. 14. An example of the eye pattern used for tracking.

3.3 Novel View Generation

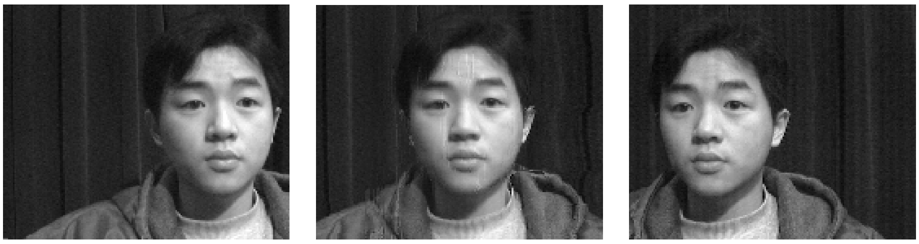
After pattern tracking and disparity maps adjustment are performed, we have two adjusted disparity maps for each stereo view. Next, we apply a linear morphing function to generate the desired view using the two adjusted disparity maps. The details of the algorithm can be found in [7, 18]. The viewing direction of the generated view can be controlled by a viewpoint parameter.

4. EXPERIMENTAL RESULTS

Our experiments were run on a Pentium III 933 MHz PC with 256MB RAM. The size of the input images was 176 × 144. The average time of disparity estimation was 6.499 sec, while the average time of disparity adjustment was 0.02 sec. In real applications, disparity estimation is required only for the first frame. Once a pair of disparity maps is obtained in the initial stage, the adjustment of the disparity and the following rending task can be done in real-time. Figs. 15 and 16 show some experimental results obtained by using the method proposed in this paper. In Figs. 15 and 16, the left and right columns show the captured stereo views. The generated views that preserve eye contact are shown in the middle column in both figures. In Fig. 15, we can see that the result of generated views look very realistic. Notice that if the speaker's motion includes a large rotation, as shown in Figs. 16 (d) and 16 (e), some noise can be observed in the generated view, especially near the boundary of the speaker. This phenomenon occurred because we did not consider the rotation information while the speaker was moving. In general, it is not easy to accurately estimate the rotation of the speaker's motion by simply tracking the speaker's eyes. We are currently working on tracking the speaker's motion by using a triangle consisting of the speaker's eyes and nose. Nevertheless, the experimental results demonstrate that the proposed real-time software method is very promising for practical applications.

5. CONCLUSIONS

In this paper, we have presented a real-time software method for preserving eye contact in video conferencing, which generates the desired view of the local participant from a stereo view captured by two cameras mounted on both sides of the monitor. The time-consuming disparity estimation procedure needs to be performed only once at the initialization stage, in order to obtain a pair of reference disparity maps. Given the reference disparity maps, our system tracks the eyes of the local participant and uses the tracking results to modify the reference disparity maps. The disparity-based view morphing procedure then utilizes the modified disparity maps to generate the desired

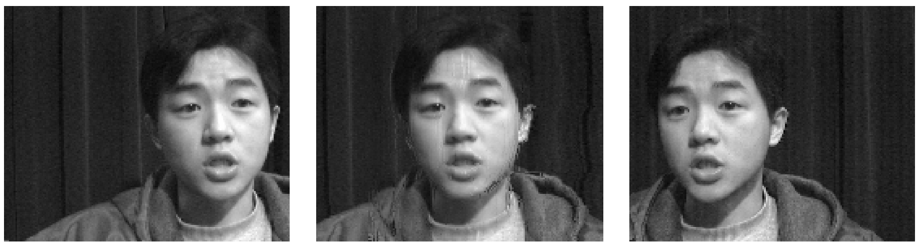


(a)

(b)

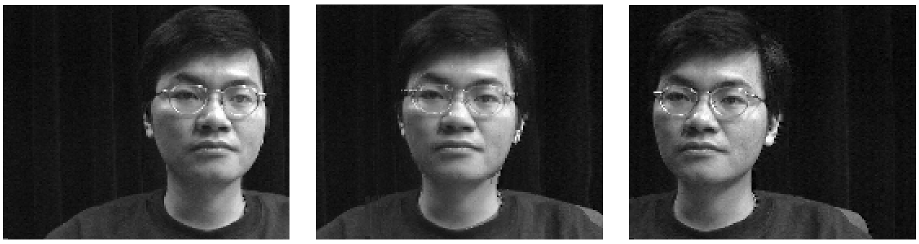
(c)

(d)

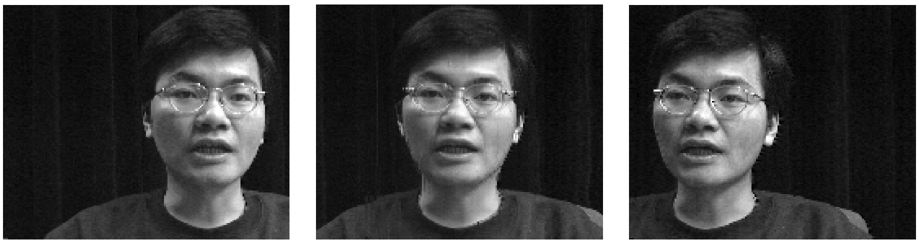


(e)

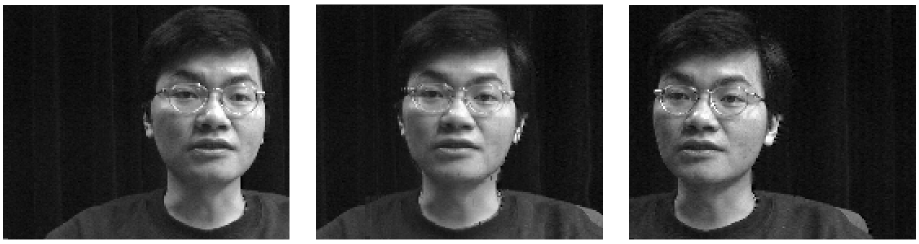
Fig. 15. Some examples of experimental results obtained by using the proposed method. The left column shows the original left image sequence, and the right column shows the original right image sequence. The generated views that preserve eye contact are shown in the middle column.



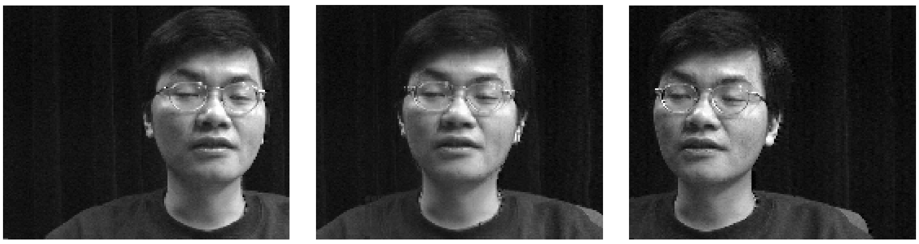
(a)



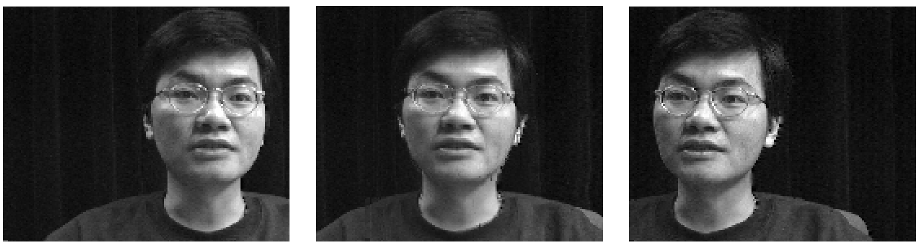
(b)



(c)



(d)



(e)

Fig. 16. Another set of example results obtained by using the proposed method. The left column shows the original left image sequence, and the right column shows the original right image sequence. The generated views that preserve eye contact are shown in the middle column.

view that preserves eye contact. Our experiments have demonstrated that the eye contact problem can indeed be solved *in real-time* by using our software approach. This is the first time a software approach has been shown to be practical for preserving eye contact in video conferencing.

REFERENCES

1. M. Accame, F. G. B. D. Natale, and D. D. Giusto, "Hierarchical block matching for disparity estimation in stereo sequences," in *Proceedings of International Conference on Image Processing* (*ICIP '95*), Vol. 2, 1995, pp. 374-377.
2. M. Bierling, "Displacement estimation by hierarchical block-matching," in *Proceedings of SPIE Conference on Visual Communication and Image Processing*, Vol. 1001, 1998, pp. 942-951.
3. N. Grammalidis and M. G. Strintzis, "Disparity and occlusion estimation for multiview image sequences using dynamic programming," in *Proceedings of International Conference on Image Processing* (*ICIP '96*), 1996, pp. 337-340.
4. N. Grammalidis and M. G. Strintzis, "Disparity and occlusion in multiocular systems and their coding for the communication of multiview image sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, 1998, pp. 328-344.
5. H. C. Huang, S. H. Nain, Y. P. Hung, and T. Cheng, "Disparity-based view morphing – a new technique for image-based rendering," in *Proceedings ACM Symposium on Virtual Reality Software and Technology* (*VRST '98*), 1998, pp. 9-16.
6. T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: theory and experiment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, 1994, pp. 920-932.
7. C. C. Kao, *Video Object Segmentation and Disparity Estimation for Novel View Generation*, Master Thesis, Computer Science and Information Engineering, National Taiwan University, 2000.
8. M. Kuriki, H. Arai, S. Sakai, S. Nakamura, H. Nakajima, S. Suyama, K. Uehira, and N. Hagiwara "Display and image capture apparatus which enables eye contact," in *U.S. Patent Documents*, US5317405, 1994.
9. S. Z. Li, *Markov Random Field Modeling in Computer Vision*, The Springer-Verlag, Tokeyo, 1995.
10. S. H. McNelley, et al., "Image blocking teleconferencing eye contact terminal," in *U.S. Patent Documents*, US5777665, 1998.
11. K. O. Mersereau, "Eye contact video telephony," in *U.S. Patent Documents*, US5666155, 1997.
12. T. Moritsu and M. Kato, "Disparity mapping technique and fast rendering technique for image morphing," *IEICE Transactions on Information and Systems*, Vol. E83-D, 2000, pp. 275-282.
13. T. J. Nelson *et al.*, "Eye contact apparatus for video conferencing," in *U.S. Patent Documents*, US5117285, 1992.
14. Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*,

Vol. 7, 1985, pp. 139-154.
15. M. Ott, J. P. Lewis, and I. Cox; "Teleconferencing eye contact using a virtual camera," *INTERACT '93 and CHI '93 Conference Companion on Human Factors in Computing Systems*, 1993, pp. 109-110.
16. L. Robert and R. Deriche, "Dense depth map construction using a multiscale regularization approach which preserves discontinuities," in *Proceedings of International Workshop Stereoscopic and Three Dimensional Imaging* (*IWS3DI*), 1995, pp. 229-234.
17. S. Roy and I. J. Cox, "A maximum-flow formulation of the n-camera stereo correspondence problem," in *Proceedings of the 6th International Conference on Computer Vision* (*ICCV '98*), 1998, pp. 492-499.
18. S. W. Shih, Y. P. Hung, and W. S. Lin, "Calibration of an active binocular head", *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, Vol. 28, 1998, pp. 426-442.
19. R. Skerjanc and J. Liu, "A three camera approach for calculating disparity and synthesizing intermediate pictures," *Signal Processing: Image Communication*, Vol. 4, 1991, pp. 55-64.
20. L. S. Smoot, "Teleconferencing terminal with camera behind display screen," in *U.S. Patent Document*s, US4928301, 1990.
21. D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 413-423.
22. C. Tomasi and R. Manduchi, "Stereo matching as a nearest-neightbor problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, 1998, pp. 333-340.
23. C. J. Tsai and A. K. Katsaggelos, "Dense disparity estimation with a divide-and-conquer disparity space image technique," *IEEE Transactions on Multimedia*, Vol. 1, 1999, pp. 18-29.
24. T. Werner, R. D. Hersch, and V. Hlavac, "Rendering real-world objects using view interpolation," in *Proceedings of the 5th International Conference on Computer Vision* (*ICCV '95*), 1995, pp. 957-962.

Yu-Pao Tsai (蔡玉寶) received his B.S. in Computer Science from the National Chengchi University in 1993, and an M.S. in Computer and Information Science from the National Chiao Tung University in 1999. He is currently a Ph.D. student in Computer and Information Science, National Chiao Tung University. His current research interests include image-based rendering, video object segmentation, and virtual reality.

Ching-Che Kao (高敬哲) received the B.S. and M.S. degrees from the Computer Science and Information Engineering Department at National Taiwan University, Taipei, Taiwan, R.O.C. in 1998 and 2000, respectively. He is currently working at Digimax Production Corporation, the leading company in Taiwan's content industry. His research interests include Image-based rendering, Computer Vision, and Stereo System.

Yi-Ping Hung (洪一平) received his B.S. in Electrical Engineering from the National Taiwan University in 1982. He received an M.S. from the Division of Engineering, an M.S. from the Division of Applied Mathematics, and a Ph.D. from the Division of Engineering, all at Brown University, in 1987, 1988 and 1990, respectively. He is currently a professor in the Department of Computer Science and Information Engineering at the National Taiwan University. From 1990 to 2002, he was with the Institute of Information Science, Academia Sinica, Taiwan, first as an associate research fellow, and then promoted as a tenured research fellow in 1997. He served as a deputy director of the Institute of Information Science from 1996 to 1997, and received the Young Researcher Publication Award from Academia Sinica in 1997. His current research interests include computer vision, pattern recognition, image processing, virtual reality, multimedia and human-computer interface.

Zen-Chung Shih (施仁忠) was born on 10th February 1959, in Taipei, Taiwan, R.O.C. He received his B.S. degree in Computer Science from Chung Yuan Christian University in 1980, M.S. degree in 1982 and Ph.D. degree in 1985 in Computer Science from the National Tsing Hua University. Currently, he is a professor in the Department of Computer and Information Science at the National Chiao Tung University in Hsinchu. His current research interests include procedural texture synthesis, non-photorealistic rendering, global illumination, and virtual reality.