An Efficient Image-Based Telepresence System for Videoconferencing

B. J. Lei, C. Chang, and E. A. Hendriks

***Abstract*—In this paper, we describe the view representation and reconstruction module for an image-based telepresence system, based on a viewpoint-adaptation scheme and the image-based rendering technique. For real-time three-dimensional synthesis, a parallel version of our multistep view reconstruction algorithm is realized on dedicated hardware based on Trimedia digital signal processors. A new scalable representation, named middle view stereo representation (MVSR), is further constructed to be able to adapt to the bandwidth requirement of the broadcasting network. Experiments show the promising performance of the parallel realization of the multistep view reconstruction method and the power of the MVSR representation.**

***Index Terms*—Image-based rendering, real time, teleconference, three-dimensional (3-D) vision, view reconstruction.**

I. Introduction

FOR SAFETY and efficiency considerations, people have been looking for remote representations of themselves for quite some time. While invisible remote representations such as intelligent agents are used more to provide certain enhanced human capabilities such as automatic doorkeeping [1] or massive information processing [2], visible remote representations are used to protect a real person while giving this person a feeling of "being there" [3] or to communicate more efficiently without tedious physical traveling [4]. For example, in advanced teleconferencing systems, this "telepresence" can bring a real sense of touch and interactive action to people physically located apart but in a shared collaborative working environment [5]. Therefore, telepresence is a powerful and commercially desirable development. With the rapid advance of key enabling technologies such as video coding and processing techniques [6] and high-speed broad-band networking [7], telepresence is under intensive investigation to innovate applications in telemedicine [8], virtual tours [9], and teleconferencing [10], among others.

In general, telepresence has two appearances: simulated presence and virtual-real presence. Simulated presence is implemented as simplified two-dimensional (2-D) or three-dimensional (3-D) avatars [11], [12]. It is very simple and can be very efficient. However, it lacks the real sense of touch and interaction. Virtual-real presence can eliminate this problem by offering an exact or near-exact copy of the outer look of the remote participant A who is telepresented at the local site for the local viewer B.

To provide the feeling of a virtual-real presence, realistic 3-D views of A should be perceived by B in real time and with the correct perspective. To satisfy this requirement, life-size views should be presented and 3-D perception should be supported. There are three visual cues essential to 3-D perception: the motion parallax cue, the stereo depth cue, and the eye lens accommodation cue [13]. Among them, the motion parallax cue is most important for 3-D perception and more practical to be realized. In fact, it can easily be provided by a *viewpoint-adaptive system*, in which the presented view to the viewer is changed in line with his/her viewpoint [13]. With a properly adapted viewpoint, a correct perspective is also guaranteed. The European project VIRTUE (VIRtual Team User Environment) [14] just aims at implementing such a viewpoint-adaptive scheme for realizing the virtual-real presence concept within a three-party teleconference application.

In VIRTUE, there are in total six communication channels. In each channel, one remote participant is connected to one local viewer [15]. For each channel (see Fig. 1), a fixed stereo setup acquires two images at the remote site. After segmentation, the pair of stereo views, containing only the remote participant without background, is broadcast to the local site. Locally, based on the information about the stereo setup, the local display, and the *pose* (position and orientation) of the viewpoint of the local viewer, these two views are used by 3-D analysis and synthesis to reconstruct a novel view ("telepresence") of the remote participant that is adapted to the current local viewpoint [15].

Two major problems in the above process are the real-time requirement for 3-D analysis and synthesis and the limited bandwidth for broadcasting (see Fig. 1). In this paper, we explore possible solutions to these two problems. To meet the real-time requirement of the 3-D synthesis, we implement a parallel version of our multistep view reconstruction algorithm on a set of TriMedia DSPs (For real-time 3-D analysis, please refer to [16]). To ease the requirement of limited bandwidth, we propose a scalable stereo representation middle view stereo representation (MVSR).

The paper is organized as follows. In Section II, the concept of image-based telepresence is explained. The above two issues are further analyzed and possible solutions are introduced. In Section III, a parallelization of the multistep view reconstruction algorithm on dedicated hardware TriMedia DSPs is discussed. The performance of this implementation is evaluated. Further, in Section IV, the structure of the new representation MVSR is

Manuscript received January 13, 2003; revised October 10, 2003. This work was supported by the European project VIRTUE (VIRtual Team User Environment).

The authors are with the Information and Communication Group, Department of Mediamatics, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands (e-mail: B.J.Lei@ewi.tudelft.nl; c.chang@ewi.tudelft.nl; E.A.hendriks@ewi.tudelft.nl).

Digital Object Identifier 10.1109/TCSVT.2004.823393

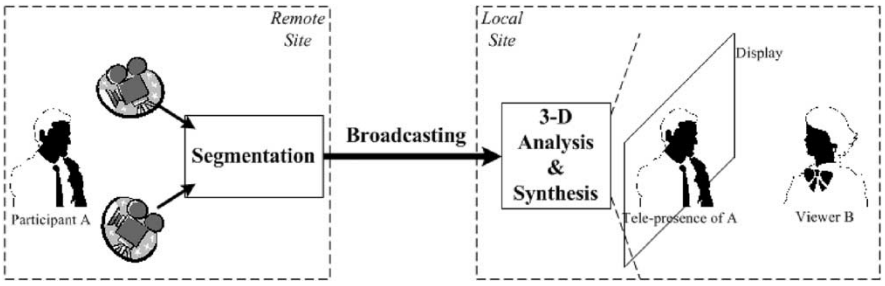

Fig. 1. Telepresence infrastructure realized in VIRTUE. For each communication channel (one viewer to one remote participant), the two cameras at the remote site provide two video streams for 3-D analysis and synthesis for the viewer in the display. The full-size remote participant A is rendered as arbitrary 2-D video objects and their synthesized looks will change in line with the head position of B. The eye-to-eye contact, normal habitual hand gesturing, and gaze awareness are expected to be maintained.

detailed. Experiments show the power of this new representation. A possible scalable architecture is also indicated. Finally, conclusions are drawn in Section V.

II. Image-Based Telepresence

Recently, the convergence of computer vision and computer graphics emerges the image based rendering (IBR) technique [17]. One big advantage of IBR is its image-size-proportional complexity (independent of the 3-D scene complexity). Thus, a system based on IBR is expected to be stable no matter how big the change of the concerned 3-D scene is. Given this advantage, it is obvious that, for the online 3-D telepresence purpose, the adaptive views of the participant are better to be constructed by IBR in real time instead of being produced from preconstructed complex 3-D models [15]. With this adoption of IBR, the telepresence developed is called *image-based telepresence*. In an image-based telepresence system, two cases can be distinguished:

Case 1) If the employed display is a normal 2-D display, then only the most important motion parallax cue is supported.

Case 2) If the display is a 3-D monitor, then the stereo depth cue can also be supported.

The techniques used behind these two cases are the same within the viewpoint-adaptive scheme. Shifting from case 1 to case 2, we only need to adapt the view obtained for the existing viewpoint (e.g., the left eye) to the added viewpoint (e.g., the right eye). Then we can get a pair of adaptive stereo viewpoints for the 3-D display.

In [15], we proposed and implemented a multistep view reconstruction algorithm (see the Appendix for an overview). We demonstrated that this algorithm is very efficient. We also indicated that this multistep method is well suited for real-time processing of CIF images on a Trimedia 133 MHz DSP. In Section III, we will discuss the implementation issues of parallelizing this algorithm for running on four Trimedia DSPs to obtain real-time processing of CCIR601 images. The motivation to this is to leave enough processing power for other parts of the telepresence system.

The multistep view reconstruction algorithm contains five steps (see Appendix):

Step 1) Stereo rectification.

Step 2) X interpolation.

(a) Original left view (b) Original right view

Fig. 2. Pair of typical stereo views from the convergent stereo setup as shown in Fig. 3. Notice the large redundancy existing between these two views.

Step 3) Y extrapolation.

Step 4) Z transfer.

Step 5) Derectification.

The input to this algorithm is a pair of stereo views, which have to be broadcasted from the remote site to the local site (see Fig. 1). However, if we take a close look at a pair of typical stereo views shown in Fig. 2, we notice that a great deal of redundancy exists between them. Therefore, for broadcasting these two views, we propose to construct a more efficient representation: MVSR. It will be shown later that this MVSR is completely a by-product of our multistep view reconstruction algorithm. Using the MVSR, the architecture used in VIRTUE (see Fig. 1) would be changed into the one shown in Fig. 3, where the three main modules are implemented, respectively, as follows.

1) **3-D Analysis**: This process contains segmentation [18], distortion correction, stereo rectification, disparity estimation [19], and MVSR construction. Distortion correction and stereo rectification can be combined for better efficiency [15].
2) **Broadcasting**: The MVSR is compressed and transmitted through a high-speed network from the remote site to the local site.
3) **3-D Synthesis**: This part implements X interpolation, Y extrapolation, Z transfer, derectification, and composition. Composition is employed to fuse the telepresence of the remote participant with a possible uniform working environment. Z transfer, derectification, and composition can be implemented as a single operation to reduce the computation load [15].

With the above task splitting, the computation load between the remote and local sites becomes well balanced.

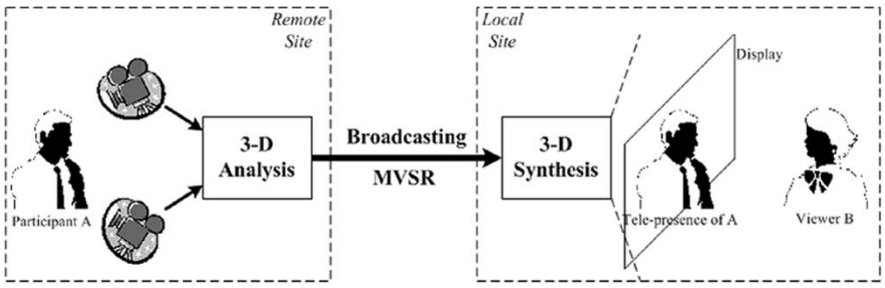

Fig. 3. Telepresence infrastructure based on MVSR.

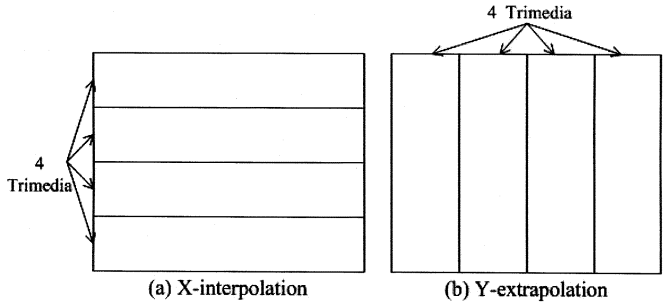

Fig. 4. Possible parallel realization of X interpolation and Y extrapolation in case four TriMedia DSPs are available. For the X interpolation, the processed views are split into four quarters in the **row** direction. While for the Y extrapolation, the processed views are split into four quarters in the **column** direction.

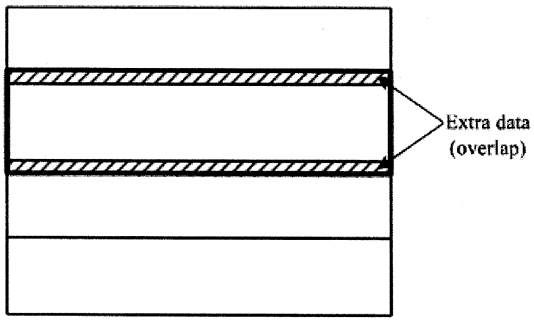

Fig. 5. Under option (a) in Fig. 4, to avoid interaction between neighboring TriMedias in the Y extrapolation, a bit more than one quarter of the data should be allocated to each TriMedia. For example, all of the data in the bold rectangle in the above group should be processed on the second TriMedia instead of only the second quarter. The extra data allocated is called *overlap*.

This MVSR structure will be discussed in detail together with its construction method in Section IV.

III. Parallel-View Reconstruction

Our multistep algorithm is feasible for real-time processing (25 fps) on a standard PC [15]. However, since other processing has to be done as well (e.g., disparity estimation, head tracking, and network handling), the parallel processing capability has to be utilized. Below, we describe the parallelization of this algorithm on four TriMedia DSPs, which are the dedicated hardware allocated to the view reconstruction in VIRTUE.

In the multistep algorithm, as indicated in [15], the rectification operation can be combined with the distortion correction. The Z transfer and the derectification steps can be put into the composition stage. Therefore, below we only need to consider how to execute the X interpolation and Y extrapolation operations in parallel. Fortunately, in X interpolation, the view is processed row by row, while in Y extrapolation the view is processed column by column independently. Therefore, if four TriMedia DSPs are available, they can be parallelized separately, as shown in Fig. 4.

However, due to the PCI bandwidth limitation and because the transpose operation of images is computationally expensive, it is impossible to switch between these two splitting methods shown in Fig. 4. Because images are organized in rows in the computer, to speed up the processing by using the most recent cached information [20], it is better to adopt the splitting option (a) in Fig. 4. As a result of this choice, X interpolation can be executed independently on the four image quarters. However, the Y extrapolation process would involve interactions between each pair of neighboring quarters. Therefore, to guarantee independence among these four TriMedias, some extra data besides the image quarter should be allocated to each TriMedia. For example, in Fig. 5, all data within the bold rectangle should be processed in the second TriMedia instead of only the second quarter. The same holds for the other TriMedias.

Still, to overcome the PCI bandwidth limitation, the data distribution scheme has to be designed carefully.

A. Data Flow

Each VIRTUE station consists of two PCs (an analysis PC and a synthesis PC) with several TriMedia PCI cards (see Fig. 6). In the 3-D synthesis module, each VS (view synthesis) calculation, containing X interpolation and Y extrapolation and realized on one TriMedia, requires the following four inputs: L (left view), R (right view), L/R (left-to-right disparity map), and R/L (right-to-left disparity map). All of these inputs are provided by a TriMedia PCI card (containing four TriMedia DSPs) in the analysis PC. After the VS calculation, each TriMedia sends its result to the host PC through the PCI bus. The compositor, running on the host PC, fuses the reconstructed views into a 3-D virtual world. It also takes care of merging the quarter frames. This whole data transformation process is shown in Fig. 7.

B. Hardware Consideration

One major limitation of the TriMedia is that one TriMedia chip can only accept one video-in stream at a time. Also, the PCI bandwidth in both the analysis PC and the synthesis PC is limited.

In Fig. 7, all data shared among the four VS TriMedia DSPs have to go through the PCI bus (see Fig. 6). Because the PCI

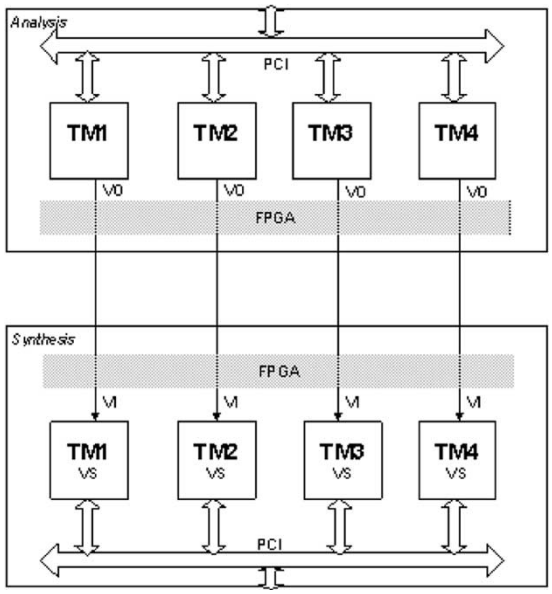

Fig. 6. Hardware configuration employed in VIRTUE. One analysis PC and one synthesis PC are used, for the purpose of 3-D analysis and synthesis, respectively. Incoming data from the network go through the analysis PC into the synthesis PC.

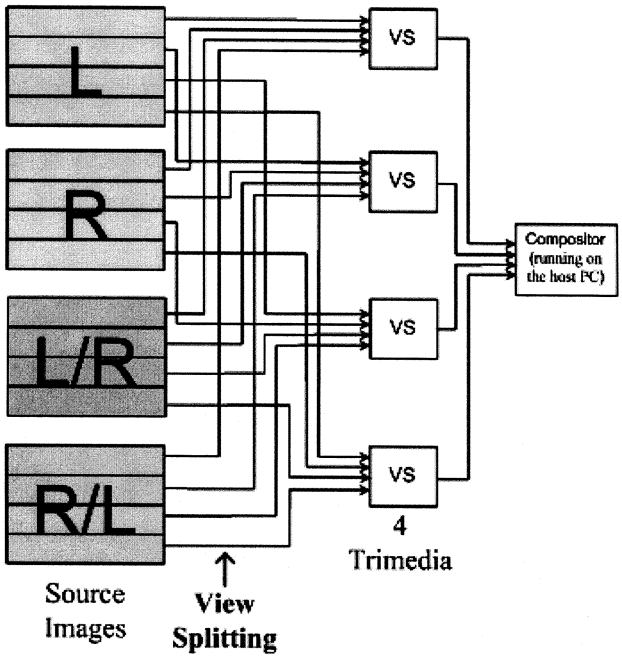

Fig. 7. Data flow required in the parallel-view reconstruction. The view reconstruction is realized in parallel as four VS modules. Each module gets a quarter of four inputs : L (left view), R (right view), L/R (left-to-right disparity map), and R/L (right-to-left disparity map). The reconstructed four quarters of the views are sent to the compositor for final processing.

bandwidth is limited, the amount of transmitted data is also limited. Experiments with the TriMedia PCI bus show that the usable bandwidth of the PCI bus is approximately 100 MBytes/s.

The results of the VS calculations are RGBA images, which have to be sent to the host PC. The bandwidth needed for transferring VS images to the compositor is approximately $720 \times 576 \times 4 \times 25 = 41$ MB/s. This means that, on the PCI bus of a TriMedia board, 41 MB/s is already used for transferring data

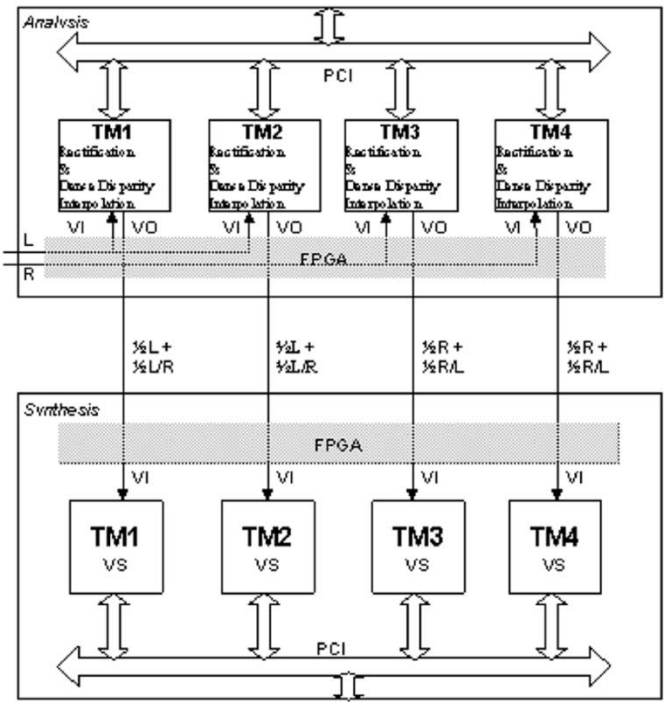

Fig. 8. Data transferring mechanism implemented in VIRTUE with the aim of fulfilling the PCI bandwidth requirement.

from the TriMedia DSPs to the host PC. So $100-41 = 59$ MB/s is left for data sharing among the TriMedia DSPs.

To fulfill this bandwidth limitation, and to minimize the computation and bandwidth burden on the analysis PC at the same time, we optimized the data transfer as shown in Fig. 8. In this mechanism, the upper four TriMedia DSPs are in the analysis PC and perform rectification and dense disparity interpolation. The first analysis TriMedia (TM) sends out the first half of the left rectified view together with the first half of the L/R disparity map to the first VS TM. The second analysis TM sends out the second half of the left rectified view and the second half of the L/R disparity map to the second VS TM. The third analysis TM sends the first half of the right rectified view together with the first half of the R/L disparity map to the third VS TM. Finally, the fourth analysis TM sends the second half of the right rectified view together with the second half of the R/L disparity map to the fourth VS TM. Note that L and R are in YUV422 format, while each pixel in L/R and R/L is of the "Byte" type. The additional PCI bandwidth requirement in the synthesis PC is

$$2\times\frac{1}{4}L+2\times\frac{1}{4}R+2\times\frac{1}{4}\frac{L}{R}+2\times\frac{1}{4}\frac{R}{L}=10+10+5+5=30\ \frac{\text{MB}}{\text{s}}$$

which fulfills the limitation given above. Only a simple multiplexing of two half images in the analysis PC is needed.

C. Overlap

As discussed above, due to the horizontal view splitting [see Fig. 4(a)], each VS TM may need some extra data from neighbor quarters for the Y extrapolation. The extra data needed for this purpose is called the *overlap*. The amount of overlap is related to the amount of the movement of the viewpoint. An adaptive strategy could be to first precalculate this amount dynamically

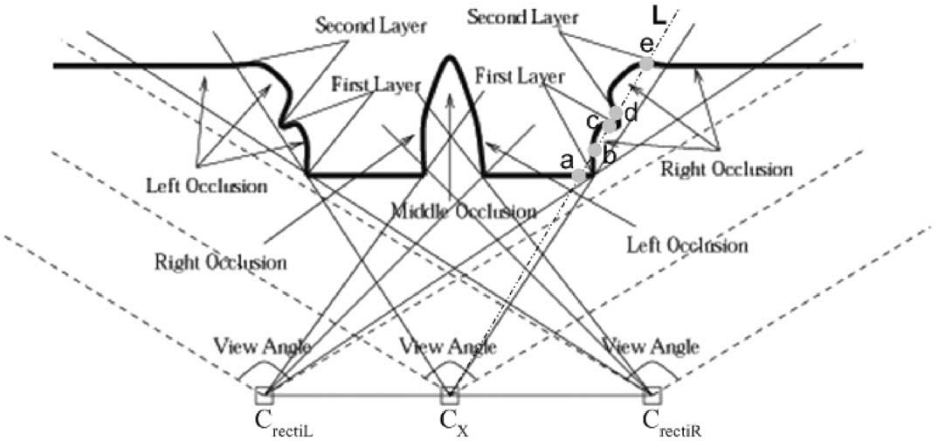

Fig. 9. Viewing at a 3-D scene from three different cameras. Bold curves indicate the scene surface from the viewpoints of the three cameras.

from the on-the-fly viewpoint and then to share only the extra data needed. However, this dynamic strategy would complicate the data transfer and the PCI bandwidth allocation schemes too much and thus would result in an even more expensive processing load. Therefore, for the current VIRTUE implementation, we just fixate the amount of overlap. This worked well with all experiments we have done so far.

D. Experiments

The geometric validity and visual quality of our multistep view reconstruction algorithm have been verified in [15]. Therefore, here we only report on the speed and final visual quality of this parallel implementation.

1) Speed: In the VIRTUE project, a processor board was developed on which 4 133-MHz TriMedia DSPs are integrated. For each view reconstruction, one such board is available. The size of the view to be reconstructed is a CCIR601 (720×576) YUV 422 image. Experiments show that, using one board, X interpolation and Y extrapolation on one frame, including the necessary data transfer process, cost $40 + -10$ ms. So, with a proper modular design of the system, a near real-time telepresence can be guaranteed.

2) Visual Quality: To show the visual performance of this algorithm, it was integrated into the VIRTUE system, where a virtual 3-D background was composed with the reconstructed telepresence. For comparison, a 2-D view of another person, which does not change in line with the viewpoint was put next to the adaptive view. This combined scene was observed from different viewpoints and at several different distances. Several such examples are shown in Figs. 20 and 21. As can be seen in these figures, a 3-D feeling can be obtained from the viewing of the left participant, but not from that of the right participant. This contrast clearly shows the viewpoint-adaptive capability of the current view reconstruction algorithm.

IV. MVSR

The MVSR is meant for encoding stereo views in an efficient and seamless way within our image-based telepresence system.

A. Available Information Types in a 3-D Scene

If we observe an arbitrary 3-D scene from the viewpoint of the left, right, and "middle" positions on the baseline, we can distinguish four types of viewed information (see Fig. 9).

1) **"Complete 3-D info"**: The information that can be viewed in both C_L and C_R.
2) **Left Occlusion**: The parts of the view that are visible in the left camera C_L but not in the right camera C_R.
3) **Right Occlusion**: The parts of the view that are visible in the right camera C_R but not in the left camera C_L.
4) **Middle Occlusion**: The parts of the view that are only visible in the "middle" camera C_X but not in both C_L and C_R.

Most "complete 3-D info" can be viewed not only in C_L and C_R, but also in C_X. The left (right) view, generated from C_L (C_R), is the "complete 3-D info" plus the left (right) occlusion parts. The "middle" view, produced by C_X, consists of most "complete 3-D info" plus the middle occlusion and parts of the left and right occlusions. If we send the original pair of two views, this "complete 3-D info" is in fact sent twice. Therefore, we need to compress the stereo data.

B. Stereo Compression

Quite a lot of work on compressing stereo data has been done. In general, there are two approaches from the view reconstruction point of view:

- **Passive compression**: Like the motion-compensated video compression scheme, this approach encodes the stereo data by a disparity compensated mechanism. One of the images is first encoded. Then the residue between its disparity-corrected version and the second image is transform coded [21]. Although this approach uses the stereo geometry to constrain the disparity search space, it does not attempt to use the recovered disparity data to approximate the true geometry of the scene but solely to reduce the information in the residue image as far as possible. Therefore, the view reconstruction can only be done after the original pair of images is recovered from the compressed data. This means that the computational burden at the view reconstruction side is increased.

- **Active compression**: This approach organizes all available views of the 3-D scene in a compact way and is specially designed to facilitate the later view reconstruction process. Examples of such compact form include image-based objects [22], layered depth image (LDI) [23], LDI tree [24], and multivalued representation [25]. However, this approach either requires a careful data-acquisition scheme [22], [25] or a deliberated representation construction procedure [23], [24]. This means that it favors the rendering part but complicates the data acquisition (compression) process.

Our MVSR attempts to balance the data acquisition (compression) and the view reconstruction. It does this by utilizing the disparity-compensation idea and by simplifying the data structure and geometry employed in LDI [23] at the same time.

In the following, we will first introduce the structure and the formation of this representation. Then we shall discuss how this representation can be used effectively for the view-reconstruction process.

C. MVSR Stucture

When telepresence is constructed as in the VIRTUE system, the scene object (participant) falls completely in the viewing volumes of both C_L and C_R. However, as the viewpoint moves from the left to the "middle" along the baseline, parts of the left occlusion will disappear while middle occlusions will become visible. On the other hand, when the viewpoint moves from the right to the "middle" along the baseline, parts of the right occlusion will disappear while middle occlusions again should become visible. This means that, if we want to fully recover the left and right views from the interpolated "middle" view, in addition to the "middle" view, these left and right occlusions and possibly a very small part of "complete 3-D info" must also be stored (which are occluded in the "middle" view). Whereas, if C_X is situated between C_L and C_R, under the ordering constraint [26], no "complete 3-D info" is occluded in the "middle" view.[1]

Because C_L, C_R, and C_X all stay on the baseline $\mathbf{b}$, here we only need to consider scene surfaces that are toward $\mathbf{b}$. For example, in Fig. 9, for C_X, for an arbitrary projection line $\mathbf{L}$, we would have information about surface points a, c and e but not about points b and d. Therefore, in the MVSR for $\mathbf{L}$, we only need to consider a, c, and e but not b and d. This at the same time means that the position of the final virtual viewpoint is largely constrained by this stereo setup. For example, we cannot look from the back of the scene object.

So, besides the visible layer ("middle" view), we should construct some extra information, named the *hidden layer*, in advance. Left occlusions in the left view, right occlusions in the right view, and a possible small part of "complete 3-D info," which are all not visible at the "middle" viewpoint, will become part of the hidden layer in the MVSR. There may exist not only one, but several hidden layers (e.g., two hidden layers in Fig. 9). If the "middle" view content becomes transparent, then the first hidden layer will become visible. If, further, the first hidden layer becomes transparent, then the second hidden layer will become visible, and so on.

The MVSR therefore contains three essential parts:

1) **"Middle" View**: This view captures most parts of the scene we are interested in.
2) **Hidden layers**: This part is in fact a view complementing the middle view to be able to reconstruct the original left and right views from the "middle" view perfectly without losing any information.
3) **Middle Disparity Map**: A disparity map based on the middle view that reflects the 3-D information of the scene.

When the baseline is too large or there are many occlusions that cannot be seen from the "middle" viewpoint, we have to record all of the hidden layers to recover the scene. When the baseline is not too large or the world scene is not too complex, we only need take into account the first layer for visual continuity. When the baseline is relatively small, we even may completely neglect the hidden-layer information to trade off the virtual view quality for the bandwidth. Currently for the VIRTUE setup [15], as the baseline is not too large, we only consider one hidden layer.

In the above, we did not record the disparity information for the hidden layers. One implicit assumption made here is that, for each horizontal segment (in each row) on the hidden layer, all pixels have the same disparity value as the lower one of the two disparity values of the two pixels in the "middle" view that correspond to the two ends of this segment. This is an idea that can produce stable and good results in view reconstruction, as analyzed in [27].

Of course, when multiple hidden layers are needed, a multibaseline configuration should be employed [28]. In this case, either the disparity values of all of the hidden layers except for the last one should be kept or a scalable structure can be constructed as that shown in Section IV-A.

D. MVSR Construction

To get both the left and right occlusion information, we need both left-to-right and right-to-left disparity maps. For occluded areas, the disparity values can be interpolated from their two ends. However, it is preferable that these values are assigned just the lower disparity value of their two ends [27]. If segmentation information is known, the best choice would be extrapolation within each segment, as we did in the VIRTUE project [19].

For constructing the stereo representation, first we synthesize the "middle" view from the left view together with the left-to-right disparity map, while the overwritten "middle" view content is sent to the hidden layer. Then we do the same thing from right to "middle". Finally we combine it in some way (e.g., in a disparity-gradient-weighted way [29]) into a integrated unit (see Fig. 10 as an example). This process is exactly the same as the X interpolation [15].

Thus, considering only one hidden layer, we get three sequences from a pair of stereo sequences. The "middle" view sequence can be compressed by, e.g., a standard MPEG encoder. The "middle" disparity sequence can first be downsampled to an acceptable level and then encoded in the same way as

[1]Assume points A and B can be viewed by both C_L and C_R. If B is occluded by A in the "middle" view, and C_X situated between C_L and C_R, then it can easily be shown that A and B will appear in different order in the view of C_L as that in the view of C_R. Thus, the ordering constraint is violated.

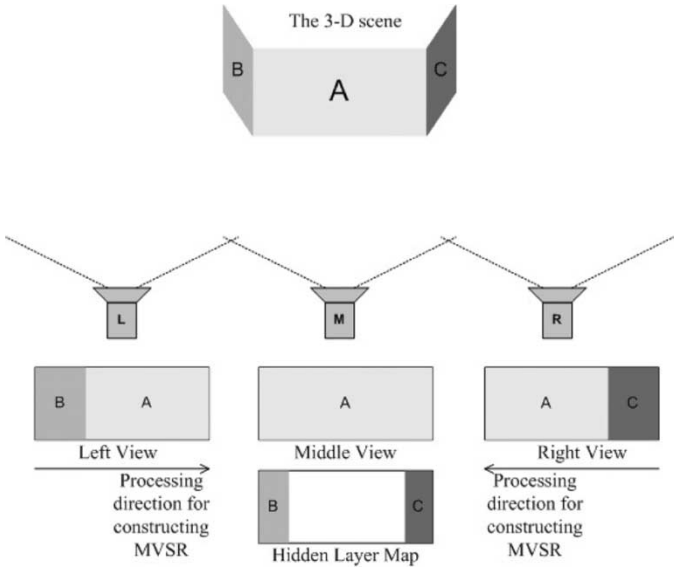

Fig. 10. Example of constructing the MVSR.

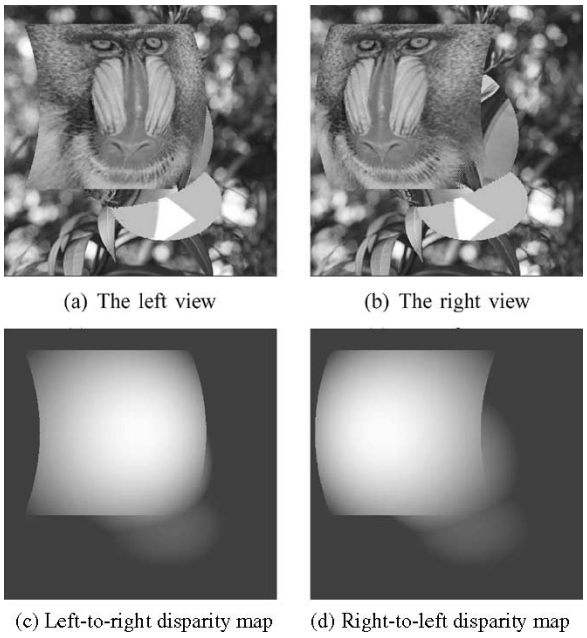

Fig. 11. Synthetic stereo pair. (Note: the values in the disparity maps are the real disparities plus a zero-offset 128).

the "middle" view sequence. The hidden-layer sequence can be transform encoded to explore the correlation between pixels in the hidden layer.

E. View Reconstruction From MVSR

With the broadcasted MVSR, arbitrary views along the baseline can be directly reconstructed. Mainly we extrapolate by forward mapping [15] from the "middle" view to the desired view. If a hole appears, we locate its two ends and determine which end has the lower disparity value. The pixel in the "middle" view that is transformed into this pixel is then located. From the corresponding position of this pixel in the hidden layer, we search for hidden information. Thereafter, we proceed as follow.

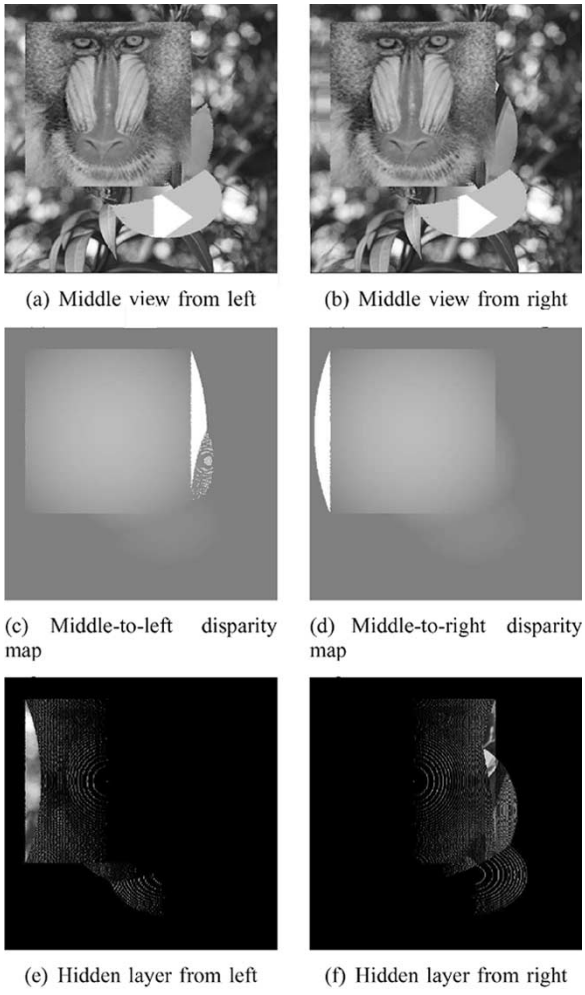

Fig. 12. Construction intermediate results for the synthetic stereo pair in Fig. 11.

- If there is hidden information, either the same length as the hole of information is filled in the hole (if the length of hole is smaller than or equal to that of the hidden information) or the hidden information is scaled to be filled into the hole (if the length of hole is larger than that of the hidden information).
- If there is no hidden information, the hole is simply filled in by linear interpolation or by using the elliptical weighted average filter [30].

This process realizes the X interpolation operation [15] in the 3-D synthesis module. In it, another hidden layer can be constructed to solve partially the disocclusion problem that may be encountered in the later Y extrapolation process [31].

F. "Middle" View Position

Note that the "middle" view employed in MVSR is not necessarily located exactly in the middle point between the right and left views. We can either select a view that contains the most important information (e.g., the view that can provide good eye contact in our teleconference system) as the "middle" view in our stereo representation or, if we know the desired X-interpolated view position in advance, we can then use this position as the "middle" point. By doing this, the X-interpolation operation

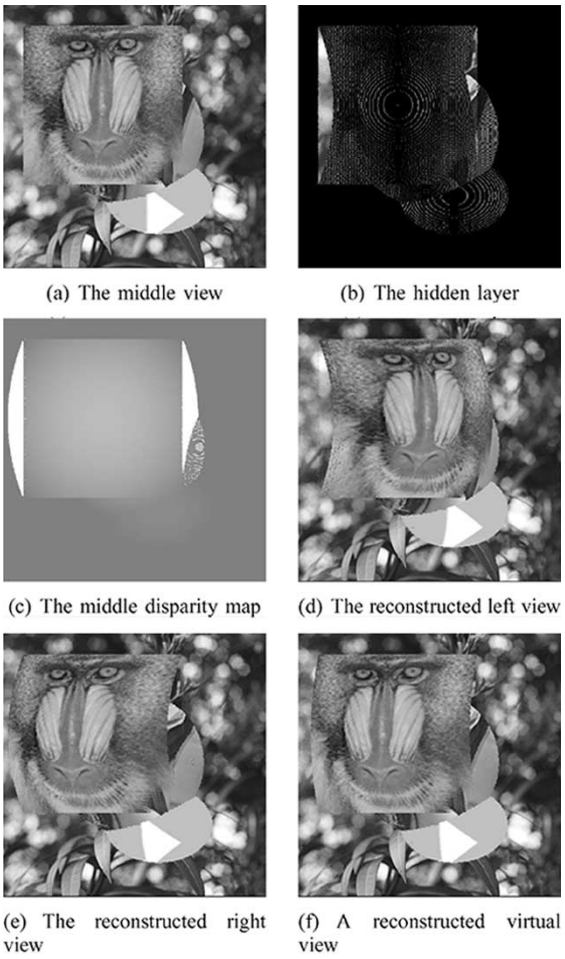

Fig. 13. Constructed MVSR and the reconstructed original stereo pair together with a synthesized virtual view.

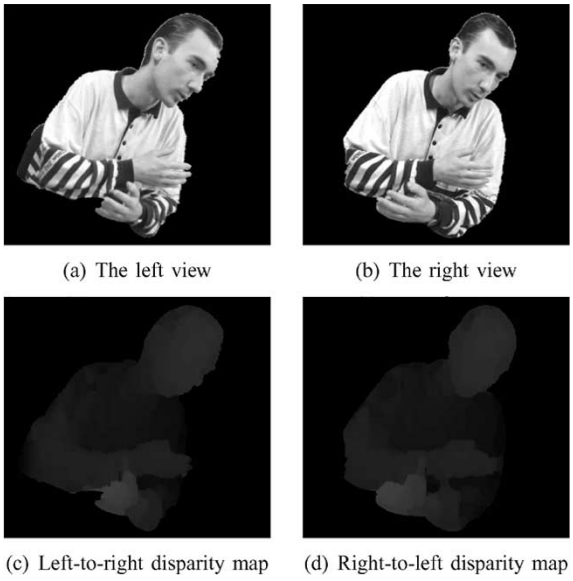

Fig. 14. Typical rectified stereo views coming from the VIRTUE setup.

at the local site can be neglected, saving further the computation time.

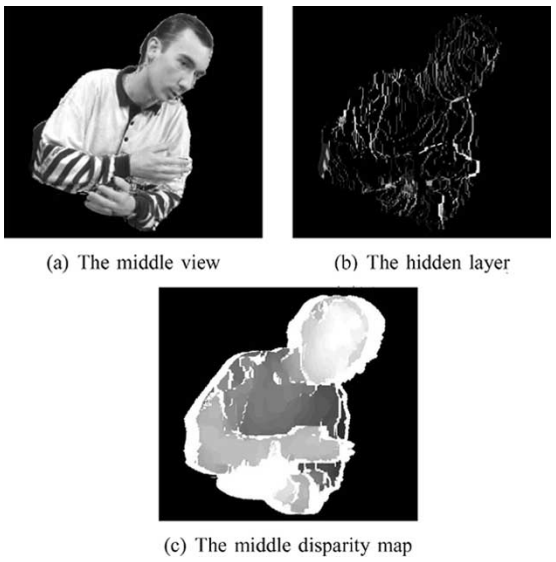

Fig. 15. Constructed MVSR for the stereo pair shown in Fig. 14. The middle disparity map is enhanced for better visibility.

Fig. 16. Reconstructed left views based on two different sources.

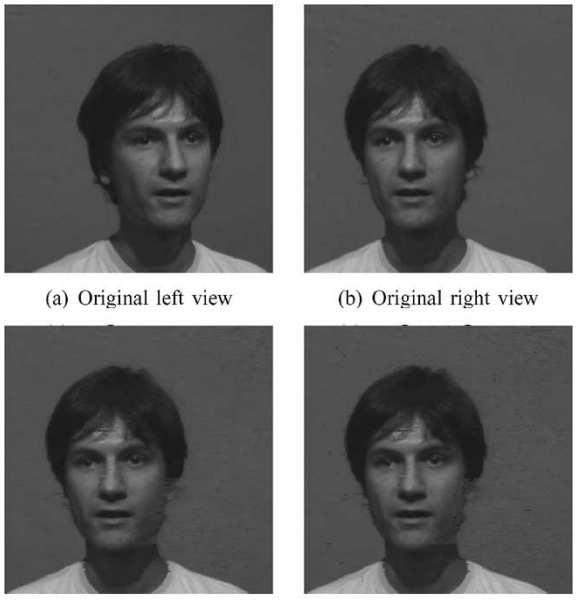

Fig. 17. Stereo views with relatively small baseline where the hidden layer may safely be neglected.

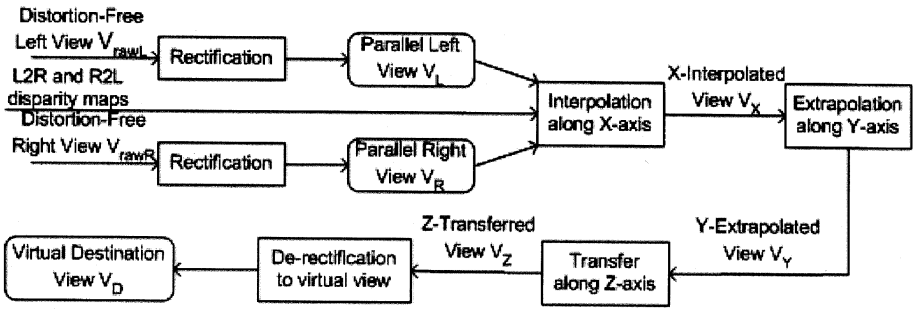

Fig. 18. View transformation framework. Together, multiple separate steps eliminate three major differences between the final novel view V_D and the two original views V_{rawL} and V_{rawR}: 1) photometric differences, such as focal length, aspect ratio, etc.; 2) position in 3-D space (x, y, and z); and 3) orientation.

G. Experiments

In all experiments below, for simplicity, the "middle" view was located exactly in the middle point and only one hidden layer was considered.

To illustrate the construction and reconstruction process via MVSR, we first show an experiment on a synthetic stereo pair together with true disparity maps (see Fig. 11). The intermediate results are shown in Fig. 12. Fig. 13 presents the MVSR reconstruction.

Note that there is some high-frequency information in the hidden layer. This is mainly due to the rounding error, because we use integer computation except for the linear interpolation used for filling of sampling gaps.

In addition to the synthetic views, we also did experiments on test sequences of the VIRTUE system. A pair of rectified stereo views coming from the VIRTUE setup is shown in Fig. 14. The constructed MVSR is presented in Fig. 15. The reconstructed left view from MVSR and that reconstructed from only the middle view are compared in Fig. 16. As can be seen in Fig. 16, in this case the hidden layer is important for high-quality reconstruction of the virtual view.

In Fig. 17, we show a case where the hidden layer may not be necessary. The reconstructed right views with and without the hidden-layer information are nearly the same. This means that, in this case, the hidden layer does not improve the results much. Thus, we may safely neglect it.

H. MVSR Advantages

One of the biggest advantages of this representation compared with other stereo compression methods is that the computation loads of the data acquisition (compression) and the view reconstruction are well balanced. A virtual view at an arbitrary viewpoint can be extrapolated from this representation in the same way as we construct MVSR. Also, the "middle" view already compensates the lighting difference between the left and right views. Therefore, the adapted view looks more consistent to the viewer. Second, it has lower requirements on the transfer bandwidth because of the low entropy of the disparity map and the low information content of the hidden layer. Third, this representation is compatible with the monovideo system and provides very good eye-contact information in teleconferencing systems such as VIRTUE.

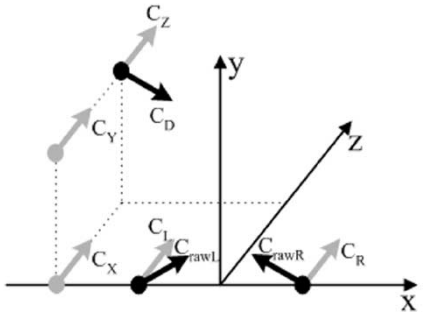

Fig. 19. Iillustration of the possible camera configurations involved in the multistep view synthesis process. The direction of each arrow indicates the orientation of the represented camera.

I. Scalable MVSR

In the above, we only discussed the MVSR structure in the stereo configuration. MVSR can also be employed in a multicamera setup. In this case, multiple MVSRs can be nested. For example, suppose there are four cameras, labeled C_1, C_2, C_3, and C_4, respectively. A structure MVSR_1 can first be constructed for C_1 and C_2. Another one, MVSR_2, can also be built for C_3 and C_4. Suppose V_{M1} is the middle view in MVSR_1 and V_{M2} is that in MVSR_2. Then a third structure MVSR_3 can be built for V_{M1} and V_{M2}. Instead of MVSR_1 and MVSR_2, only MVSR_3, disparity maps and hidden layers in MVSR_1 and MVSR_2 need to be transmitted.

V. Conclusion

Image-based telepresence is stable with the help of the image-size-proportional property of the IBR technique. Based on this idea, real-time view reconstruction, which is essential for online tele-interaction, can be achieved with simple hardware (e.g., TriMedia) as shown in Section III-D. The visual quality of the image-based telepresence is sufficient (cf. Figs. 20 and 21). Three-dimensional perception can be obtained directly from the dynamically adapted view together with the 3-D virtual world. More than this, an efficient representation such as MVSR can get rid of the redundancy embedded in the broadcast views to increase the efficiency of the network bandwidth utilized.

The purpose of this paper is twofold. First, we have shown and discussed how to realize a real-time full-sized CCIR601

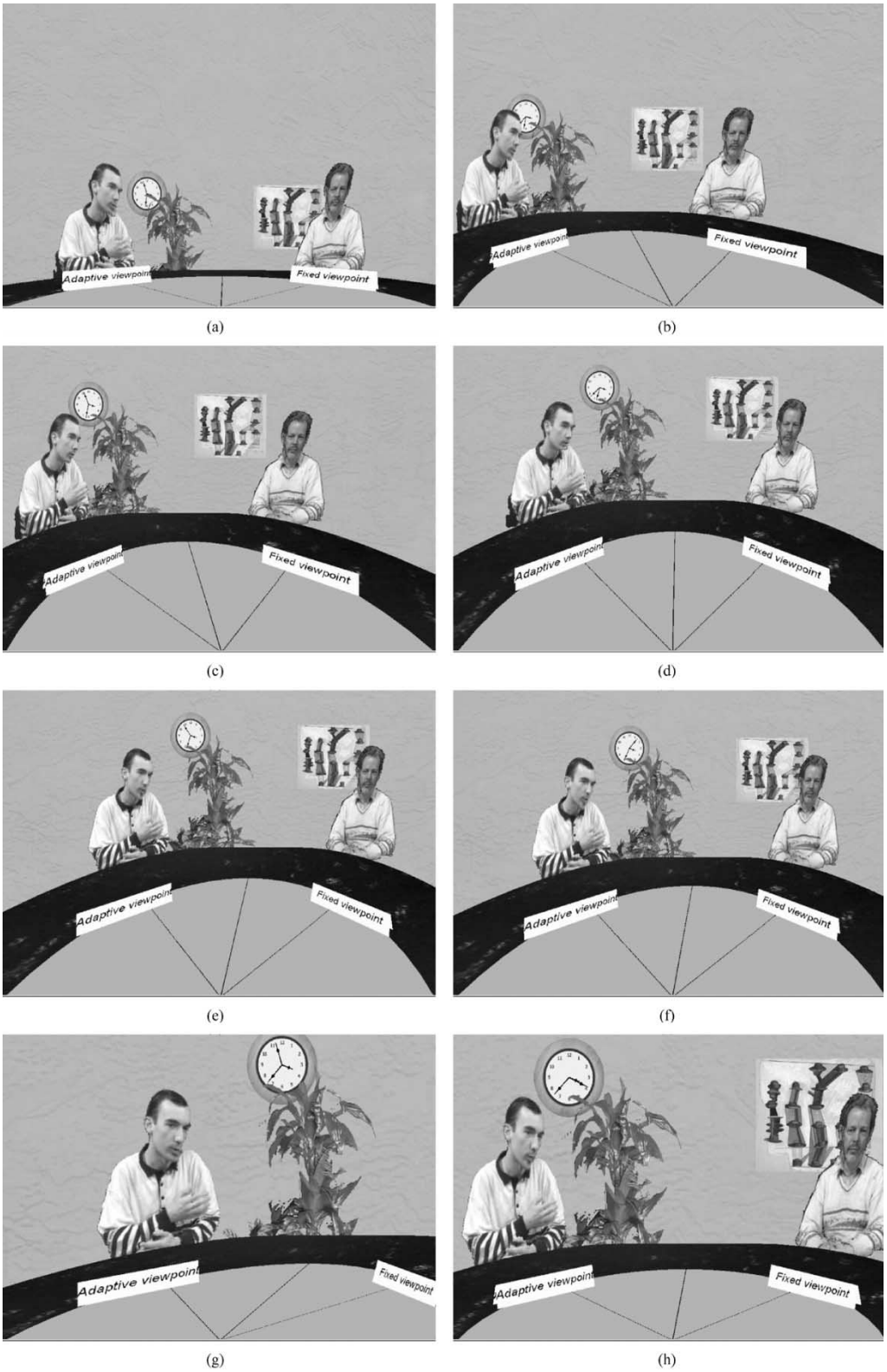

Fig. 20. Example scenario of the VIRTUE system observed from several different viewpoints. The contrast between the adapted view of the left participant and the static view of the right participant clearly shows the viewpoint adaptive capability of the current view reconstruction algorithm.

adaptive view reconstruction by parallelizing the view reconstruction algorithm on four 133-MHz Trimedia DSPs. We have indicated that the speed is in fact limited by the necessary data transfer to and between the DSPs. Second, we have proposed an MVSR that is scalable and from which an arbitrary virtual view can be efficiently reconstructed. The visual quality of the final reconstructed telepresence views is good. The scalability makes a tradeoff between quality and possible bandwidth.

Our current discussion is based on a stereo setup and for a specific telepresence system. We only explored the spatial re-

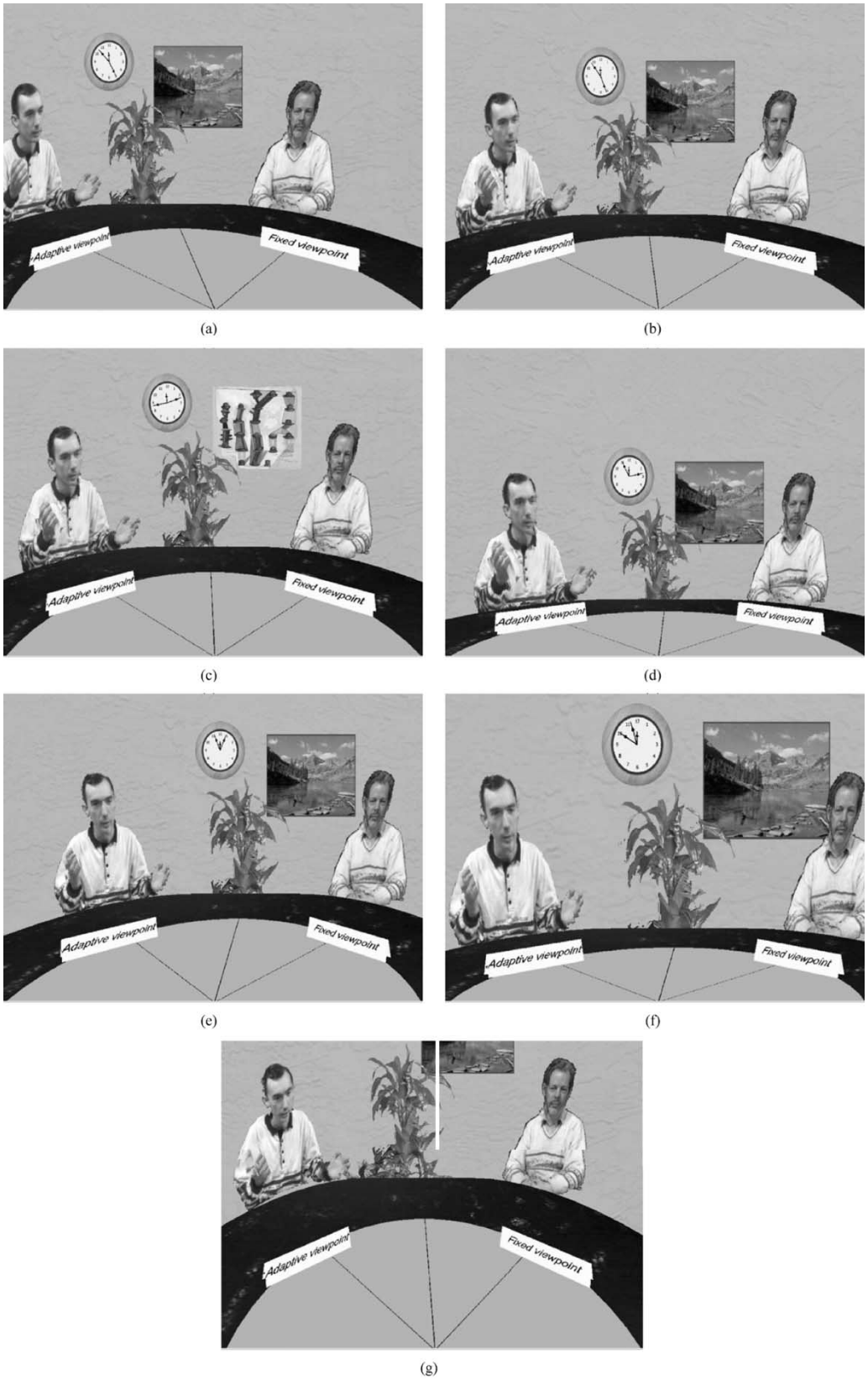

Fig. 21. Another example scenario similar to that in Fig. 20. Notice the relative positions between both hands and the body. In (f), the face looks stretched. The cause to this is that the error in the disparity maps is exaggerated by a big viewpoint deviation. This tells us: 1) for a fixed setup, the virtual viewpoint that can be reconstructed is largely constrained and 2) if we want a desired range of virtual viewpoints, the camera setup should be modified correspondingly.

dundancy embedded in the stereo views; for the temporal redundancy, we passively used an MPEG encoder. However, as investigated in [32], the temporal continuity property of motion can also be utilized to facilitate the view reconstruction process.

Therefore, a combined approach that considers both spatial and temporal redundancy would be more promising. Also, as already mentioned, the number of hidden layers needed is directly related to the complexity of the 3-D scene and the configuration of the camera setup. By digging out this intrinsic relationship, researchers might find another solution to the problem of how to effectively sample a 3-D scene by images [33].

APPENDIX

The objective of view reconstruction is to reconstruct a virtual view V_D for a virtual camera C_D from a pair of stereo views, V_{rawL} and V_{rawR}, which are generated from two cameras, C_{rawL} and C_{rawR}, respectively.

As a starting point for the following discussion, without loss of generality, the World Coordinate System (WCS) can be selected such that

$$\begin{aligned}\mathbf{t}_{\mathrm{crawL}} &= [1 \quad 0 \quad 0]^T \\ \mathbf{t}_{\mathrm{crawR}} &= [-1 \quad 0 \quad 0]^T \\ \mathbf{t}_{cD} &= [x_{cD} \quad y_{cD} \quad z_{cD}]^T\end{aligned}$$

where $\mathbf{t}_{\mathrm{crawL}}$, $\mathbf{t}_{\mathrm{crawR}}$, and $\mathbf{t}_{cD}$ are the position vectors of C_{rawL}, C_{rawR}, and C_D, respectively. This means that the x axis of the WCS lies on the baseline $\mathbf{b}$ of C_{rawL} and C_{rawR} and points from C_{rawR} to C_{rawL}. The origin of the WCS is at the middle point on $\mathbf{b}$, that is, the unit of distance is $b/2$ (b is the length of $\mathbf{b}$).

In the most general case, the multistep view-reconstruction process can be divided into five steps (see Fig. 18).

1) Rectification: Transforming the stereo views V_{rawL} and V_{rawR} into a pair of new views V_L and V_R, respectively: the two virtual cameras C_L and C_R that generate these two new views are parallel to each other and share the same image plane. This process is known as stereo rectification **[34]** and is intended to eliminate the photometric differences and orientation differences between the two source cameras to simplify the correspondence estimation into a one-dimensional (1-D) search problem along the scan line and at the same time to provide parallel processing possibilities for later steps.

2) $\mathbf{X}$ interpolation:: Given necessary disparity information, the two parallel views V_L and V_R are combined by interpolation or extrapolation **[35]** to produce another parallel view V_X. The corresponding camera C_X is located at $[x_{cD} \quad 0 \quad 0]$ with the same rotation and intrinsic parameters as C_L and C_R. The y coordinate of each pixel remains the same, while the x coordinate is transformed by $x_p^X = x_p^L + (1 - x_{cD}/2)d_p^{LR}$ and/or (in case of occlusion) $x_p^X = x_p^R + (1 + x_{cD}/2)d_p^{RL}$, where x_p^* is the x coordinate of pixel p^* in view V_* ($* = X, L, R$). p^X, p^L and p^R are projections of the same 3-D point. d_p^{LR} and d_p^{RL} are disparities of p^L and p^R, respectively, where $d_p^{LR} = x_p^R - x_p^L$ and $d_p^{RL} = x_p^L - x_p^R$. Note that, in the case of occlusion, either p^L or p^R is not available. Through this step, the difference in the x position with the final view V_D is eliminated.

3) $\mathbf{Y}$ extrapolation: The X-interpolated view V_X is extrapolated **[36]** by shifting the pixels in the Y direction to produce the view V_Y, which comes from a virtual camera C_Y located at $[x_{cD} \quad y_{cD} \quad 0]$ with the same rotation and intrinsic parameters as C_X. In this process, the x coordinate of each pixel remains the same while the y coordinate is transformed by $y_p^Y = y_p^X - y_{cD} \cdot (s_x/s_y) \cdot d_p^X$, where y_p^* is the y coordinate of pixel p^* in view V_* ($* = X, Y$). d_p^X is the disparity of p^X, where $d_p^X = d_p^{LR}/2$ or (in case of occlusion) $d_p^X = -d_p^{RL}/2$. Through this step, the difference in the y position with the final view V_D is eliminated.

4) $\mathbf{Z}$ transfer: The Y-extrapolated view V_Y is transferred along the Z direction to generate a closer or more distant look V_Z. The corresponding camera C_Z is located at $[x_{cD} \quad y_{cD} \quad z_{cD}]$ with the same rotation and intrinsic parameters as C_Y. Both the x and y coordinates of each pixel are transformed in a similar manner as the X interpolation and Y extrapolation. However, the dimension of the view is maintained **[15]**. The z-position difference to the final view V_D is eliminated. It should be noted that, for different application situations, this Z transfer step could be simplified or modified in different ways for better computational performance **[15]**

5) Derectification: The Z-transferred view V_Z is rotated and scaled to get the final view V_D

In Fig. 19, an illustration is given of the possible camera configurations involved in the multistep view-reconstruction process.

ACKNOWLEDGMENT

The authors are thankful to the partners HHI, BT, Sony UK, and HWU in the VIRTUE consortium for valuable discussions and relevant contributions.

REFERENCES

[1] R. Stockton and R. Sukthankar, "Argus: The digital doorman," *IEEE Intell. Syst.*, vol. 16, pp. 14–19, Mar./Apr. 2001.
[2] N. Azarmi, S. Case, T. Ohtani, and M. Thint, "Enhancing e-communities with agent-based systems," *Computer*, vol. 34, no. 7, pp. 64–69, July 2001.
[3] N. Negroponte, *Being Digital*: Vintage Books, 1996.
[4] *Telepresence*, P. Sheppard and G. Walker, Eds., Kluwer, Norwell, MA, 1999.
[5] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs, "The office of the future: A unified approach to image-based modeling and spatially immersive displays," in *Proc. SIGGRAPH'98*, 1998, pp. 179–188.
[6] Y. Wang, J. Ostermann, and Y. Zhang, *Video Processing and Communications*. Upper Saddle River, NJ: Prentice-Hall, 2002.
[7] E. Carne, *Telecommunications Primer: Data, Voice, and Video Communications*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.
[8] L. Burke and B. Weill, *Information Technology for the Health Professions*. Upper Saddle River, NJ: Prentice-Hall, 2000.
[9] Y. Onoe, K. Yamazawa, H. Takemura, and N. Yokoya, "Telepresence by real- time view-dependent image generation from omnidirectional video streams," *Comput. Vis. Image Understanding*, vol. 71, no. 2, pp. 154–165, 1998.
[10] L. Xu, B. Lei, and E. Hendriks, "Computer vision for 3d visualization and telepresence collaborative working environment," *BT Tech. J.*, pp. 64–74, January 2002.
[11] A. Mortlock, D. Machin, S. McConnell, and P. Sheppard, "Virtual conferencing," in *Telepresence*, P. Sheppard and G. Walker, Eds. Boston, MA: Kluwer, 1999, pp. 208–226.
[12] *Deformable Avatars*, vol. 196, IFIP Conference Proceedings, Kluwer, 2001.
[13] P. Redert, "Multi-viewpoint systems for 3-d visual communication," Ph.D. dissertation, Delft Univ. of Technol., Delft, The Netherlands, 2000.
[14] (2000–2003) Eur. IST Project IST-1999-10 044. VIRTUE. [Online]. Available: http://www3.btwebworld.com/virtue/
[15] B. Lei and E. Hendriks, "Real-time multi-step view reconstruction for a virtual teleconference system," *Proc. EURASIP J. Appl. Signal Process.*, vol. 2002, no. 10, pp. 1067–1088, Oct. 2002.

[16] O. Schreer, C. Fehn, N. Brandenburg, M. Karl, and P. Kauff, "Fast disparity estimator for real-time video processing using a hybrid block- and pixel-recursive matching technique," in *Proc. Picture Coding Symp.*, Seoul, Korea, 2001, pp. 405–408.
[17] H. Shum and S. Kang, "A review of image-based rendering techniques," in *Proc. IEEE/SPIE Visual Communications and Image Processing (VCIP) 2000*, Perth, Australia, June 2000, pp. 2–13.
[18] O. Schreer, I. Feldmann, U. Goelz, and P. Kauff, "Fast and robust shadow detection in videoconference applications," in *Proc. VIPromCom 2002, 4th EURASIP IEEE Int. Symp. Video Processing and Multimedia Communications*, Zadar, Croatia, June 2002, pp. 371–376.
[19] O. Schreer, N. Brandenburg, S. Askar, and P. Kauff, "Hybrid recursive matching and segmentation-based postprocessing in real-time immersive video conferencing," in *Proc. VMV'2001*, Stuttgart, Germany, Nov. 2001, pp. 383–390.
[20] A. G. LaMarca, "Caches and algorithms," Ph.D. dissertation, Univ. of Washington, Seattle, 1996.
[21] H. Aydinoğlu and M. Hayes, "Stereo image-coding: A projection approach," *IEEE Trans. Image Processing*, vol. 7, pp. 506–516, Apr. 1998.
[22] M. M. Oliveira and G. Bishop, "Image-based objects," in *Proc. Symp. Interactive 3D Graphics*, 1999, pp. 191–198.
[23] J. Shade, S. Gortler, L. He, and R. Szeliski, "Layered depth images," in *Proc. SIGGRAPH'98*, Orlando, FL, July 1998, pp. 231–242.
[24] C. Chang, G. Bishop, and A. Lastra, "LDI tree: A hierarchical representation for image-based rendering," in *Proc. SIGGRAPH'99*, Los Angeles, CA, Aug. 1999, pp. 291–298.
[25] N. Chang and A. Zakhor, "A multivalued representation for view synthesis," in *Proc. ICIP'99*, Kobe, Japan, Oct. 1999, pp. 505–509.
[26] A. Bobick and S. Intille, "Large occlusion stereo," *Int. J. Comput. Vis.*, vol. 33, no. 3, pp. 181–200, Sept. 1999.
[27] D. Scharstein and R. Szeliski, "Stereo matching with nonlinear diffusion," *Int. J. Comput. Vis.*, vol. 28, no. 2, pp. 155–174, July 1998.
[28] J. Webb, "Implementation and performance of fast parallel multi-baseline stereo vision," in *Proc. Computer Architectures for Machine Perception*, 1993, pp. 232–240.
[29] P. Redert, E. Hendriks, and J. Biemond, "Synthesis of multi viewpoint images at nonintermediate positions," in *Proc. ICASSP'97*, vol. IV, Los Alamitos, CA, 1997, pp. 2749–2752.
[30] N. Greene and P. S. Heckbert, "Creating raster omnimax images from multiple perspective views using the elliptical weighted average filter," *IEEE Computer Graphics Applicat.*, vol. 6, pp. 21–27, June 1986.
[31] M. M. Oliveira, "Relief Texture Mapping," Ph.D., University of North Carolina, 2000.
[32] W. Mark, L. McMillan, and G. Bishop, "Post-rendering 3d warping," in *Proc. Symp. I3D Graphics*, 1997, pp. 7–16.
[33] C. Zhang and T. Chen, "A Survey on Image-Based Rendering – Representation, Sampling and Compression," Carnegie Mellon Univ., AMP03-03, 2003.
[34] R. Hartley, "Theory and practice of projective rectification," *Int. J. Comput. Vis.*, vol. 35, no. 2, pp. 115–127, Nov. 1999.
[35] S. Seitz and C. Dyer, "Physically-valid view synthesis by image interpolation," in *Proc. Workshop on Representation of Visual Scenes*, Cambridge, MA, 1995, pp. 18–25.
[36] D. Scharstein, *View Synthesis Using Stereo Vision*. Berlin, Germany: Springer Verlag, 1999, vol. 1583, Lecture Notes in Computer Science (LNCS).

B. J. Lei received the B.Sc. degree in computer software and the M.Sc. degree in parallel network computing from Xi'an Jiaotong University, Xi'an, China, in 1995 and 1998, respectively, and the Ph.D. degree from the Technical University of Delft, Delft, The Netherlands, in 2003.

He is currently involved in low-level image processing, three-dimensional imaging, and computer vision, and he enjoys developing practical multimedia applications.

C. Chang received the M.Sc. degree in electrical engineering from the Delft University of Technology, Delft, The Netherlands, in 1998.

In 1999, he joined the Information and Communication Theory Group, Delft University of Technology, as a System Engineer where he was responsible for the hardware implementation of the view synthesis with the European Virtue project. His interest is in hardware/software combinations for real-time solutions.

E. A. Hendriks received the M.Sc. and Ph.D. degrees from the University of Utrecht, Utrecht, The Netherlands, in 1983 and 1987, respectively, both in physics.

In 1987, he joined the Electrical Engineering Faculty, Delft University of Technology, Delft, as an Assistant Professor. In 1994, he became a member of the Information and Communication Theory Group, Electrical Engineering Faculty, Delft University of Technology, and since 1997 he has been the head of the Computer Vision section of this group as an Associate Professor. His interests are in computer vision, low-level image processing, image segmentation, stereoscopic and three-dimensional imaging, motion and disparity estimation, gesture recognition, structure from motion/disparity/silhouette, and real-time algorithms for computer vision applications.