Eurographics Symposium on Virtual Environments (2004)
S. Coquillart, M. Göbel (Editors)

A Tele-immersive System Based On Binocular View Interpolation

Pierre Boulanger, Martha Benitez and Winston Wong

Department of Computing Science
University of Alberta
2-21 Athabasca Hall, Edmonton, Alberta, Canada, T6G 2E8

Abstract

The main idea behind tele-immersive environment is to create an immersive virtual environment that connect people across networks and enable them to interact not only with each other, but also with various other forms of shared digital data (video, 3D models, images, text, etc.). Tele-immersive environments may eventually replace current video and telephone conferencing, and enable for a better and more intuitive way to communicate between people and computer systems. To accomplish this, participants to a meeting has to be represented digitally with a high degree of accuracy in order to keep a sense of immersion. Tele-immersive environments should have the same "feel" as a real meeting. Interactions among people should be natural. In other to create such a system, we need to solve the key problem of how to create in real-time new views from a fixed network of cameras that will correspond to new viewpoints. We also need to do this for two virtual cameras corresponding to the inter-ocular distance of each participant. In this paper, we will describe a new binocular view interpolation method based on a re-projection technique using calibrated cameras. We will discuss the various aspects of this new algorithm and of the hardware systems necessary to perform these operations in real-time. We will also present early experimental results illustrating the various advantages of this algorithm.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism: Virtual Reality

1. Introduction

Communications technology is constantly changing the way in which people interact, both for business and personally. With the advent of such technologies as video conferencing, web conferencing and high speed networks, communicating has taken a huge step forward from email and telephone. It is now possible to communicate via real-time video and audio, bringing a new dimension to the disembodied voice of the telephone. With ever growing improvements to networking technology and increases in processing power and display capability, the next generation of communication tools are just around the corner. The shared virtual environment provides a possible successor for the traditional video conference. Shared virtual environments is an emerging concept that leverages technologies found in computer networking and the fields of virtual reality (VR) or virtual environments (VE's). VR or VE's, can be defined as a broad term used to describe an immersive, interactive, computer generated 3D environment and its associated interface and display technologies. What distinguishes virtual environments from 3D graphic environments is the idea of 'immersion', meaning the user is totally absorbed inside the virtual world while outside stimuli are minimized. This is usually accomplished using large stereoscopic displays.

Shared virtual environments attempt to create an immersive, shared, 3D world that connects people across networks and enables them to interact not only with each other, but with various other forms of digital data e.g. 3D models, video, images, text and sounds. Seamlessly integrating the various digital data in a shared VE allows an increased level of awareness over traditional video conferencing. The goal is to produce an environment that enables a more realistic, flexible and intuitive way to communicate. To facilitate natural communication, scenes and particularly participants must be represented as realistically as possible. The term ’avatar’ is generally given to a representation of a user inside a virtual environment. Interactions among participants take place through their avatars, so having realistic avatars able to represent the full range of human expressions in real-time is very important. To create a faithful 3D digital reproduction of real world scenes and people, shared VE’s must capture real world environments and events and place them into the shared virtual world. This differs from traditional VR which has concerned itself with creating synthetic digital environments that mimic the real world. The capturing process takes place through real-time sensors, the most common of which are cameras used to capture video. Cameras provide the opportunity to view the real world in a photo-realistic real-time manner, making it an ideal choice for representing scenes and avatars in a shared VE.

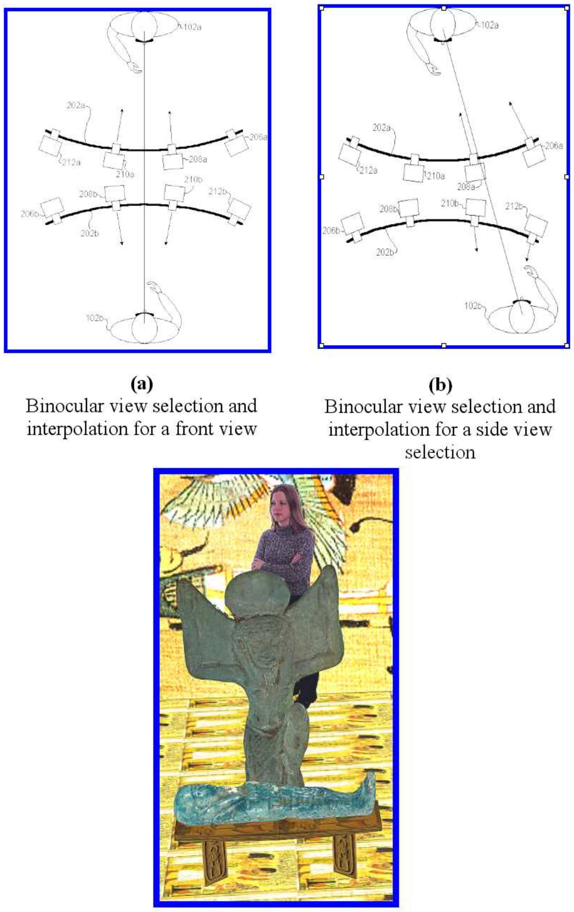
1.1. Proposed Tele-Immersive System

The proposed system described in this paper use data from a position tracking sub-system to locate the participants (102a, 102b), and then determines which two cameras (208a and 210a in Figure 1a) that most closely approximate a view of participant (102a) from the perspective of participant (102b).

Using the tracking sub-system, the system then selects those two cameras to supply video images of participant (102a) to participant (102b). The system similarly selects cameras 208b and 210b to supply video images of participant 102b to participant 102a. The system then separates the image of each participant 102a, 102b from the background that appears in the respective video images. This is done using an improved version of the PFINDER algorithm [27] that perform foreground /background segmentation using a background mask and some stereo information supplied by the two closest video cameras.

The system then transforms these respective video image pairs to create a stereo pair of video images separated by a nominal inter-ocular spacing of participant 102b. A view interpolation algorithm similar to the one described in [3] [24] is used and improved for this context. Each transformed video pair is then transmitted to the participants (102a), (102b) and incorporated into the respective participant’s view of the virtual meeting on a polygon located in the virtual world and display using a stereo texture technique.

As participants move around at their respective locations, the system tracks their position and select appropriate camera pairs. In the example, shown in Figure 1b, when participant 102b moves forward and to the right of the position showed in Figure 1a, the system select cameras 210b and 212b to capture views of participant 102b. The system likewise select cameras 208a and 206a for providing the most appropriate perspective for capturing views of participant 102a to be supplied to participant 102b. The position information related to each participant is also used to process the captured audio of each participant’s voice, in order to reproduce the sound of each participant’s voice in the 3D space of the virtual meeting room and to locate the rendered polygon representing the participant.



(a)
Binocular view selection and interpolation for a front view

(b)
Binocular view selection and interpolation for a side view selection

(c)
Inserted stereo video texture in the virtual world

Figure 1: *Proposed tele-immersive system*

Depending on the positions of participants in a shared VE, they will see different views of the environment and its participants. If a participant were to change position or orientation their view of the scene would also change, thus requiring a new view to be rendered. Creating or synthesizing these new views of a scene is the problem of view synthesis and is the main topic of this paper. The goal is to create novel views that allow for smooth transition between of viewpoint. View synthesis raises an important issue with the use of cameras for shared VE’s. Increasing the number of cameras would allow switching of cameras to represent changing views, but this can quickly become impractical with large numbers of cameras required for smooth view transition. Another solution is to create the new views using the information provided by the existing set of camera images, this type of solution falls into the category known as image-based rendering. The view synthesis problem is closely related to the way in which scenes are represented; how new views are generated depend on the underlying scene representation. Scene representation generally falls into one of two classes: 3D model-based or image-based representations. 3D model-based approaches, where the scene and/or avatars are 3D models, make the view synthesis question simple. Since the 3D geometry is known, new views can be created by simply rendering the scene from the different viewpoints. Image-based approaches, where cameras are used to collect information on the scene or avatar, have a more difficult time with view synthesis as new views must be generated from an existing, discrete, set of images. In this paper, we assume the use of cameras to serve video to participants of a shared VE. This implies an image-based approach making the view synthesis problem nontrivial.

One of the basic issue that need to be solved for this system is the problem of real-time view interpolation that create a perfectly adapted stereo pair for each participant from the two selected cameras. Because of limited space, this paper will focus mainly on this problem. Although as important as view interpolation, the problem of background segmentation and tracking will not be described in details. In Section 1, we will review various view synthesis methods. In Section 2, we will describe the problem of view morphing in the context of creating a new stereo pair from two cameras. In Section 3, we will describe the algorithm implemented and in Section 4, some experimental results obtained so far. We will then conclude and describe future work.

2. Review of View Synthesis Methods

In this section, we review some of the work that one can find in the literature to solve the problem of creating new views from a sparse network of video cameras.

2.1. View Synthesis from 3D Reconstruction

3D reconstruction involves taking sensor, usually camera, inputs of a scene and reconstructing a 3D model. 3D reconstruction techniques have been studied extensively in computer vision. While there are promising results, reconstruction is a complex and difficult problem. Extracting 3D scene structure from images can be a very unstable process. It also suffers from being computationally expensive and sensitive to errors. Methods for 3D reconstruction generally fall into one of several categories: structure from stereo, structure from motion, active scanning, structured lighting and shape from shading. This also covers the major techniques for reconstruction, though combination are also possible e.g structure from a stereo image sequence.

The seven camera system of the National Tele-Immersion Initiative (NTII) project [15] was used in 3D reconstruction of an avatar. To aid in the capture and processing of the image data, 'imperceptible structured lighting' was used to overlay patterns on the scene which could only be seen by the synchronized cameras. Areas lacking in features would be filled in by the structured light, giving information to the reconstruction algorithms in otherwise blank areas. Updates of the avatar occurred at 2-3 times per second, using consumer level hardware. While not currently real-time, increases in computational power and improved algorithms could eventually achieve real-time performance in the coming years [15]. Although this system would only capture a single person sitting at a desk, addition of cameras would increase the allowable range of movements of a user. The '3D Room' at Carnegie Mellon University consists of 49 calibrated and synchronized cameras [12], an offshoot from the 'sea-of-cameras' approach by Fuchs et al. [7]. This configuration is able to capture any event occurring inside the room and reconstruct a 3D model. The video is processed of-line and can then be flown through interactively at a later time, a 3D digital video. A system from Zaxel Systems Inc has been created that effectively performs reconstruction in real-time. Derived from the work done in the 3D Room, a system of cameras is used to capture shape and texture and then allow real-time virtual viewpoint generation. The system uses powerful PC level hardware with a proprietary software and camera system and is designed to be set up in a booth or room type setting. The background of the room is removed automatically by the system aiding in the production of avatars. As a model-based technique, it is still dependant on scene complexity and would not generalize well to complex environments (e.g. outdoor scenes). This system has been used by [20] in an augmented reality setting to produce avatars that can be viewed from virtual viewpoints.

2.2. View Synthesis from Imaged Based Method

The image based approach to scene representation completely avoids having to create an explicit 3D model. Dubbed image-based rendering (IBR), a scene is represented by a base set of camera images. Scenes and avatars are rendered photo-realistically up to image resolution of the cameras, providing for more realism than even a high resolution 3D model. There is no reliance on scene complexity, as any 3D modelling is completely avoided, and since there is no reconstruction, no 3D information needs to be extracted from the images. Pure image-based rendering requires no knowledge of scene geometry. View synthesis is not as trivial as in the model-based approach, because of a lack of explicit 3D scene geometry, new scenes need to be synthesized from existing base image sets. The techniques used to accomplish pure image-based view synthesis rely on the plenoptic function to characterize a scene. The plenoptic function [1], is a 7D function, P, that describes intensity of light rays at every location (V_x, V_y, V_z), at every angle (θ , ϕ), for every wavelength λ, at time t.

$$p = P(\theta, \phi, \lambda, V_x, V_y, V_z, t) \qquad (1)$$

To generate a new view, a position (V_x, V_y, V_z), orientation (θ, ϕ) and time t are supplied to the plenoptic function. The plenoptic function can be thought of as providing all possible views of a scene. This implies having to capture an unrealistic amount of images to fully characterize a scene, in practice there are much fewer images available. Due to this restriction, image-based rendering can be described as attempting to generate a continuous plenoptic function given a discrete set of samples [17].

Current methods are unable to deal with a complete 7D plenoptic function. Instead, the plenoptic function is simplified based on elimination of some of the variables. Plenoptic modelling by McMillan and Bishop [17] keep the environment and lighting conditions static, thus t and λ may be dropped resulting in a 5D plenoptic function. Limiting the available viewpoints to a bounding box around an object, as done in the lumigraph [8] and lightfield [14] techniques, can further reduce the 5D plenoptic function to 4D. Perhaps the most widely known method of IBR is Apple's QuickTime VR [5], based on the simplest 2D plenoptic function, P(θ, ϕ), where the camera position is fixed (V_x, V_y, V_z) are unchanged and images are taken in a cylindrical panorama. It should be noted that these methods are based on the assumption of a static environment, that is, t is constant.

2.3. View Synthesis Using Hybrid Methods

Hybrid methods are those methods which draw from model-based and image-based representations. These are methods that do not rely solely on scene geometry or base image sets, but a combination of geometry and image information. These techniques can be considered as rendering with implicit scene geometry [28].

Using a small number of images on which geometric constraints are applied, image pixels can be reprojected to form a new image from a novel viewpoint. These types of techniques are referred to as transfer methods by the photogrammetric community [28].

There are many view synthesis methods that belong to this category and take advantage of geometric constraints obtained through computer vision techniques. One such category, called view interpolation (also called correspondence techniques), make use of image correspondence's to perform image warping and produce new views based on a small set of base views, generally a stereo image pair. Examples of view interpolation methods can be found in [2] [21] [11]. One particular variation of interest for shared VE's is view morphing that was implemented for this system.

Among the first view synthesis techniques that could be applied to real world scenes was the work of Laveau and Faugeras [13]. Working with uncalibrated images, an image-warping technique that produced perspective correct views was used in conjunction with the fundamental matrix. Five user specified corresponding point pairs were used to determine the re-projection of image points onto a virtual view. These points indirectly specified the center of projection and orientation of the image plane for the virtual camera view. This work demonstrated view synthesis was possible from weakly calibrated cameras and dense correspondence maps.

The view morphing work of Seitz and Dyer [23] use the ordering constraint along epipolar lines to avoid needing complete correspondence. Image morphing and interpolation is used to produce physically valid views along the line joining camera centers.

View morphing algorithms can be represented in a modular way, making implementation flexible and allowing for real-time considerations. For our application, we only have to view morph a stereo image pair, the following three steps are necessary:

- Image rectification is performed on the images using warping matrices computed from the calibration process;
- Source image is linearly morphed towards the target image to acquire a rectified in-between image;
- A postwarp (de-rectification) is applied to the morphed image to obtain the final virtual view;

Lei and Hendriks [16] use an interpolation technique for view synthesis that is designed for real-time use in the VIRTUE 3D teleconferencing system. It is designed in a modular way to facilitate implementation in real-time. The technique, similar to view morphing, uses two calibrated cameras and dense correspondence. Applied specifically to the use of avatars, the background is first segmented away. The following steps are followed to generate the virtual view: 1. Rectification 2. Successive interpolation in the x, y, z coordinates to bring view to the desired virtual camera position.

More recently, HP Research has developed Coliseum [4] a multiuser immersive remote teleconferencing system. In this system, five cameras are attached to each PC monitor and directed at the participant. The Coliseum system is based on the Image- Based Visual Hulls (IBVH) image-based rendering scene reconstruction technology of MIT [18]. HP researchers have shown that the IBVH method can operate at video rates from multiple camera streams hosted by a single personal computer. Coliseum enables users to share a virtual world, with acquired-image renderings of their appearance replacing the synthetic representations provided by more conventional avatar-populated virtual worlds.

The view morphing approach has the advantages of a pure IBR approach, in that modelling or extraction of explicit 3D information is avoided. At the same time, unlike a pure IBR method, it is possible to perform this in real-time on dynamic scenes and since view morphing operates on a two image set, there are none of the associated storage costs. This is an ideal solution for use in a shared virtual environment, as it should provide realistic scenes at an acceptable level of performance. In the following sections, we will discusses and outline our real-time view morphing algorithm.

3. Real-time View Morphing for Shared Virtual Environment

One limitation to existing transfer methods for view synthesis, view morphing included, is that the range of virtual views that can be generated are somewhat more limited in range than those generated by a model-based method. In the case of view morphing this is restricted to intermediate views between the two camera images. Although the two camera case is considered here, the algorithms and techniques will readily extend to multi-camera configurations allowing for an increased range of views that can be synthesized as described in the introduction. Multiple camera systems can simply be treated as a series of two-camera systems, where an appropriate pair of cameras is chosen based on the viewpoint that is to be rendered as described in the introduction.

Introduced by Seitz and Dyer [22] [23], view morphing is a view interpolation technique that allows the synthesis of transitional views from one image to another. This solves the view synthesis problem for in-between or interpolated virtual views lying along the line joining the camera centers of the original images (basis images), it is not a technique for generating arbitrary novel virtual viewpoints. This section details the view morphing algorithm that was implemented in our system.

3.1. View Morphing Algorithm

For use in shared virtual environments, we require a completely automatic method for generating virtual views. The main problem with adapting view morphing is in eliminating the need for any user input. In particular, the feature specification and correspondence aspect must be automated. This is an important step, as these features are used in the image morph and any incorrectly matched features will cause image distortions in the final view. The following algorithm works to automate the feature specification and correspondence stages from the standard view morphing algorithm.

In the following algorithm, we propose to use a minimal set of object features in the images. We will demonstrate that by detecting the contour of the participant relative to its background and some other features such as the location of the eyes and/or the nose, we can perform view interpolation accurately and efficiently. From these feature points, we then apply a feature-based stereo matching algorithm to find edge correspondences. Finally, we perform a linear morph using the feature correspondences to obtain the interpolated, in-between view.

In the proposed system, the following two processes are running in parallel:

Transmission Process:

- Initialize H323 transmission with other participants;
- Initialize the system by calibrating the cameras;
- Compute the warping matrix for each cameras using the calibration parameters including lens distortions;
- Transmit warping matrices to each participants;
- Accumulate ten frames of the background image without the participant to create two segmentation templates, one for the left camera and one for the right camera;
- Create two rectified image templates using statistics extracted from the ten background images and the warping matrices;
- Compute a dense disparity map template from the two rectified background templates using a simple correlation method along epipolar lines;
- **For each frame**
  - Digitize the left and right images;
  - Pre-warp both images using their respective image warping matrices;
  - Segment foregound/background using a technique similar to the PFINDER, i.e. Baysian classification of the pixel using their color. The background pixels are set to black;
  - Detect the largest connected component and extract its contour;
  - Match feature points along the contour as well as for the points inside the largest connected component that are classified as background.
  - Compute disparity for each of these points using a simple correlation technique along epipolar lines;
  - If the disparity of these points is significantly larger (10%) then the background disparity template change their classification to foreground;
  - Detect eyes and nose positions in the connected region using a simple template matching algorithm described in [9].
  - Create a label map that classifies pixels as foreground, background and legal feature points;
  - Compute the location of the participant relative to the common VR world using the disparity along the contour;
  - Broadcast position to other participants;
  - Encode and broadcast warped images and label map to participants using H323;
- Loop until the end of transmission;

Reception Process:

- Initialize H323 connection with remote participants;
- Read common virtual meeting room 3D model;
- For each participants read initial positions in the virtual world and their corresponding warping matrices;
- **Main rendering loop**

- **For each participant**
  - Read and decode the rectified stereo pairs and the label map;
  - Read participant location;
  - Update participant polygons to new value;
  - Establish correspondences between the features of stereo pairs using the label map;
  - Linearly interpolate corresponding edges per scanline to obtain warped source and target images, which are then cross-dissolved to obtain the new rectified interpolated stereo pairs for the receiver intraocular distance.
  - Postwarp the two new stereo pairs using the warping matrices associated to this participant.
  - Download the rectified stereo pair to the texture memory associated to the participant polygons;
- Perform scene rendering;
- Loop until the end of transmission.

At the core of view morphing is the calculation of the boundary flow [24]. Due to the aperture problem calculation of dense disparity maps can be ambiguous. However, image discontinuities corresponding to scene object boundaries can be more reliably computed. Seitz argues that this is sufficient for predicting the appearance of the object in a new view. Uniform regions will remain uniform regions in a new view, the only change will be in it's shape which is defined by it's boundaries. To ensure that uniform regions are indeed maintained through different viewpoints the ordering constraint must be satisfied and extended to all views that can be synthesized. Recall that the ordering constraint implies uniqueness, and so no occlusions should occur in the range of new views. Monotonicity limits the types of scenes that can be considered for view morphing, but the practical application of the algorithm discussed below provides fairly sensible output to violations, causing only localized distortions in new views [22].

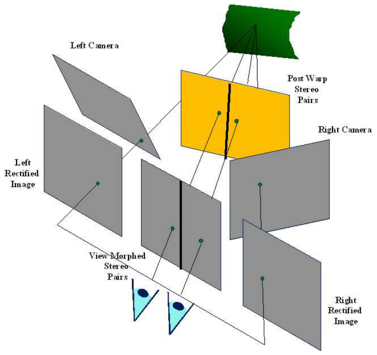


Figure 2: *Non-parallel views: Input images are rectified and an two intermediate images is generated corresponding to inter ocular distance of the participant, then these images are transformed back to the original viewpoint.*

Figure 4, describes the projection of region boundaries on two base images I_l and I_r and interpolated view I_{ls} for the left image and I_{rs} for the right image, where $s \in [0,1]$ specifying the linear transition from image I_l ($s = 0$) to I_r ($s = 1$). For the case of parallel views taken with parallel cameras of differing focal length (the simple stereo geometry), we can show an interpolated new view is a view with a projection matrix, $\mathbf{M}_s$, linearly interpolated from the projection matrices of the base images, $\mathbf{M}_l$ and $\mathbf{M}_r$. Following directly from [23], we have the following:

For simplicity camera, the left camera center C_l, is placed at the origin of the Euclidean world coordinates and right camera center C_r, is placed at $(C_x, C_y, 0)$. A point $\mathbf{P}$ is a scene with homogenous coordinates, $\mathbf{P} = [x, y, z, 1]^T$. Let $\mathbf{p}_l$ and $\mathbf{p}_r$ be the projections of $\mathbf{P}$ onto image I_l and I_r respectively. Assuming a simple pin-hole camera model, we can write the projection matrices for $\mathbf{M}_l$ and $\mathbf{M}_r$ as follows:

$$\mathbf{M}_l = \begin{bmatrix} f_l & 0 & 0 & 0 \\ 0 & f_l & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{M}_r = \begin{bmatrix} f_r & 0 & 0 & -f_r C_x \\ 0 & f_r & 0 & -f_r C_y \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

Linearly interpolating $\mathbf{p}_l$ and $\mathbf{p}_r$ gives,

$$(1-s)\mathbf{p}_l + s\,\mathbf{p}_r = \frac{(1-s)}{Z}\mathbf{M}_l\mathbf{P} + \frac{s}{Z}\mathbf{M}_r\mathbf{P} = \frac{1}{Z}\mathbf{M}_s\mathbf{P} \tag{3}$$

where the projection matrix $\mathbf{M}_s$ is define by:

$$\mathbf{M}_s = (1-s)\mathbf{M}_l + s\,\mathbf{M}_r \tag{4}$$

Thus for a new view stereo pair centered at $\mathbf{C}_s$, lying on the line $\overline{C_l C_r}$, the projection matrix for the image pair is $\mathbf{M}_{ls}$ and $\mathbf{M}_{rs}$, which represents two linear interpolations of the camera centers at position $\mathbf{C}_{ls} = (s_l\ C_x, C_y, 0)$ and $\mathbf{C}_{rs} = (s_r\ C_x, C_y, 0)$ and two focal lengths equal to $f_{sl} = (1-s_l)f_l + s_l\ f_r$ and $f_{rl} = (1-s_r)f_l + s_r\ f_r$. For obvious reasons, we make sure that the focal length of each camera is the same and equal to f_o. The interpolation parameters s_l and s_r are equal respectively to $s_l = s - IOC/2C_x$ and $s_r = s + IOC/2C_x$ where IOC is the intraocular distance of a participant.

Note that the points which are interpolated correspond to region boundaries. In this way, we can obtain the shape of objects in the interpolated view. The color information between boundaries is filled in through another interpolation, which is part of a linear morph from source image I_l to I_l. For the more general case of non-parallel views, a prewarping step is performed, which rectifies the images to the parallel view case.

3.2. Image Rectification

Image rectification simplifies the epipolar geometry in such a way as to make all epipolar lines parallel in the horizontal direction, while aligning corresponding points in the vertical direction (See Figure 3). By aligning epipolar lines horizontally, they coincide with image scanlines, thus allowing

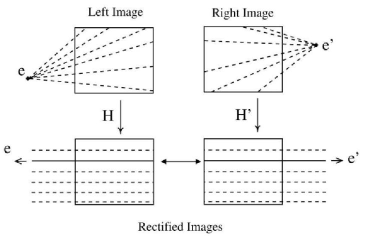


Figure 3: *Image rectification process.*

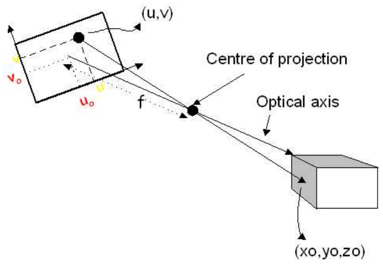


Figure 4: *Definition of the camera calibration parameters.*

algorithms to take advantage of optimal scanline techniques. This allows image warping and stereo matching algorithms to execute more efficiently. The first step in image rectifications is to correct for the effect of radial and off axis distortions created by lens abberations and alignments with the CDD sensor. These corrections are particulary important if one want to use low cost cameras such as webcam where the quality optical of the cameras construction is very poor. The following equations shows how to obtain undistorted coordinates $\mathbf{p} = (x, y)$ from the observed image coordinates $\mathbf{p}_o = (x_o, y_o)$:

$$x = x_o + (x_o - c_x)(a_2 r^2 + a_3 r^4) + \quad (5)$$
$$a_4(r^2 + 2(x_o - c_x)^2) + 2a_5(x_o - c_x)(y_o - c_y)$$
$$y = y_o + (y_o - c_y)(a_2 r^2 + a_3 r^4) + \quad (6)$$
$$a_5(r^2 + 2(y_o - c_y)^2) + 2a_4(x_o - c_x)(y_o - c_y)$$

where (a_2, a_3) compensate for radial distortions created by the lens and (a_5, a_4) compensate for the fact that the optical axis of the lens and image plane are not necessarily perpendicular.

In addition for this lens distortion compensation, We need to rectify the images in such a way that correspondences are aligned vertically and that epipoles are mapped to a point at infinity (See Figure 3). The transformation from a point $\mathbf{p}^{'} = (x^{'}, y^{'}, 1)$ on the parallel plane to a point $\mathbf{p} = (x, y, 1)$ in the image plane is defined by:

$$\mathbf{p} = \mathbf{K}\mathbf{p}^{'} \quad (7)$$

The parameter of the so called calibration matrix $\mathbf{K}$ are intrinsic parameters of the camera and are evaluated during calibration. This second rectifications process is performed by computing for the left and right images the following transportation $\mathbf{p}_l^{'} = \mathbf{K}_l^{-1}\mathbf{p}_l$ and $\mathbf{p}_r^{'} = \mathbf{K}_r^{-1}\mathbf{p}_r$

3.3. Calibration Procedure

In order to represent a real world camera, there is a set of parameters to be evaluated and defined. They are grouped into two categories:

- Intrinsic parameters
- Extrinsic parameters

3.3.1. Intrinsic Calibration

Intrinsic parameters are used to represent the internal characteristics of the cameras: the optics and hardware elements that specify the corresponding projection that generates the images. These parameters are respectively c_x and c_y, the projective center of the camera, f_x and f_y the ratio of the focal length of the camera f expressed in pixels over p_x and p_y the width and hight of the pixels, and finally w the skew factor due to non-rectangular pixels. They can be grouped in a so called calibration matrix $\mathbf{K}$ defined by:

$$\mathbf{K} = \begin{bmatrix} f/p_x & w & c_x \\ 0 & f/p_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

In addition to the projective transformations there is also the distortion parameters that need to be computed. We refer the reader to [6] for more details.

3.4. Extrinsic Calibration

These parameters correspond to the position and orientation of the camera into the world coordinate system. These are essential parameters if one want to track user in the environment. The cameras motion can be represented in matrix from as follows:

$$\mathbf{M}^{'} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ \mathbf{0}_3^T & 1 \end{bmatrix} \quad (9)$$

where $\mathbf{R}$ represent a rotation matrix and $\mathbf{t} = [t_x, t_y, t_z]^T$, translation vector.

3.4.1. Calibration Procedure

To perform calibration a network of target was shown to the cameras. Efforts was made to ensure that all the cameras see at the same time the target assembly. For this system, we used a full photogrammetric calibration program capable of computing all the intrinsic and extrinsic parameters in parallel using a bundle adjustment algorithm. Using this program, its is possible to estimate all the cameras parameters at the same time making sure that the calibration parameters are consistent from one cameras to the other. Using ShapeCapture 4.0 [25] the procedure is very fast, automatic and without any problems. It takes typically, five minutes to calibrate. This process is far superior to autocalibration process described in [19] that are slow and tend to be unstable because on non-unique solutions.

3.5. View Interpolation Algorithm

On of the fist set of view interpolations is to establish correspondence between the two images. In our implementation, we use the contour extracted from the foregound/background segmentation process and other points representing key features on the face such as the eyes or the nose.

Once we have edge correspondences, we may perform the image morph consisting of a linear interpolation and cross-dissolve. We perform the interpolation on a per scanline basis. The algorithm is the following:

- For each corresponding edge pixel pair (e_l, e_r), on a scanline, where e_l is an edge in I_l and e_r the corresponding edge in I_r.
- Linearly interpolate the edges to their new position in the interpolated view: $e_w = (1-s)e_l + s\, e_r$
- Map the regions between edges in I_l and I_r to the regions between the newly interpolated edges. Generating the left warped images I_{wl} and the right warped images I_{wr} one scanline at a time.
- Cross-dissolve the warped images to obtain the final morph I_s. This is performed by interpolating intensity, on a per pixel basis.

$$Intensity(I_s) = (1-s)Intensity(I_{wl}) + sIntensity(I_{wr})$$

The mapping step of the algorithm is implemented as a 1D forward resampling function [26]. The method works by treating incoming pixels, as being able to contribute completely or partially to an output pixel. Contributions are accumulated and then output as the final pixel. This works to deal with the under-sampling (aliasing) problems.

4. Experimental Results

As a first approximation to the more complex system described in the introduction, we have developed a prototype versions illustrated in Figure 5. It is composed of two synchronized video camera mounted on a 18 inch DTI autostereo display screen allowing to view in stereo without glasses. As illustrated in Figure 6, the two cameras are combined into one frame using an analog side by side multiplexer. This combined frame and its associated audio signal is then coded using H323 hardware codex (Vicon Vigo) and transmitted over the network to a second site where the signal is then decompressed and view morphed to adapt to the inter-ocular distance of each participant. The resulting stereo pair is then inserted into the virtual world displayed in stereo. One of the advantages of using autostereo displays is that there is no need to wear glasses that would hide the participant eyes. Our current experimental results show that even though the motion of the head is strongly limited by its working principle, these type of display device are indeed very useful. A new generation of autostereo display was released recently which do not limit as much the position of the head.

The prototype system is based on an Athlon 1.4GHz with a 64MB Nvidia Geforce4 video card running under Window XP, OpenGL is used in rendering the images. The typical speed for the reconstruction of a 320 x 240 image is around 10Hz which is sufficient considering the fact that decoding and transmission of the image over the network is approximately the same.

At the base of a good view morphing process is the selection of the proper features necessary to establish correspondence between the two images. We experimented with various features and various combinations of them. One can see in Table 1, the sum of the difference between a reference image taken by an independent camera and the view morphed image for the case with contour only, contour and eyes, contour and nose, and finally contour, eye, and nose. For this set of images only 12 line segments were needed to approximate the person's contour and the corresponding 4 segments for the eyes and nose when needed. As one can see in Figure 7a, the result of view morphing for the contour only. One can notice that even though most of the features are properly aligned there is residual blur created by the fact hat a key feature such as the nose was not included. On can see the effect in Figure 7b the effect of adding the nose location as a feature. In this implementation we detected the nose automatically using an algorithm described in [9]. Experimental results shows that this algorithm is very robust and fast. Nose tracking is to our viewpoint an excellent feature since it is rarely occluded.

Even though these preliminary results are encouraging none of the code developed during this first phase of the project was optimized for speed. For example, in the next phase we will try to minimize the number of line by segmenting the contour using a fast split and merge algorithm. A second speed-up will be to use dual processors. The first processor will be for decoding H323 and for graphic, and the second one will be reserved only for view morphing and feature extraction.

5. Conclusion

The straightforward implementation of the proposed algorithm leads to performance ratings that was implemented in a near real-time environment with fairly realistic results.

The quality of the view interpolation is heavily dependent

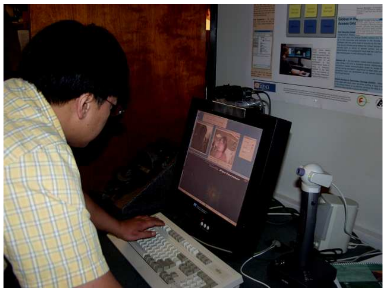


Figure 5: *Prototype tele-immersive system composed of two synchronized camera, a side-by-side multiplexer, a H323 hardware codex, and an autostereo display from DTI.*

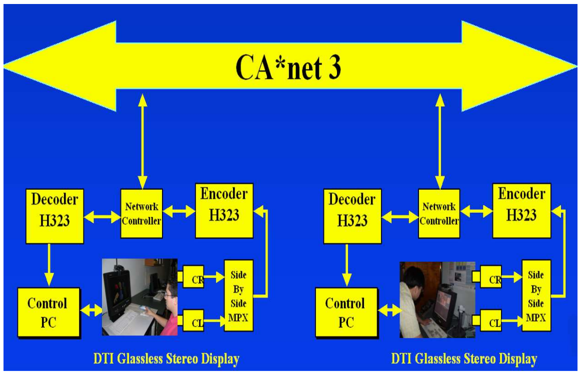


Figure 6: *Block diagram of the prototype system for peer to peer transmission.*

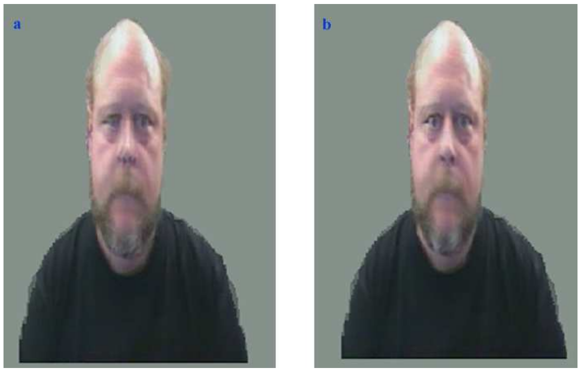


Figure 7: *View interpolation results: (a)Contour only (b)Contour and nose.*

on the steps prior to the morph, especially the type of feature used and correspondence stages. Errors in the features correspondence cause horizontal streaking distortions, which ruin the realism of the interpolated view. This restricts the scenes which can be realistically interpolated to those from which we can obtain reliable features.

While the algorithm takes care of the automation of feature detection and view generation, there remain some challenges before full adoption to a shared virtual environment. In the current configuration, the range of possible virtual views is limited to only those views which lays in-between two base images. This may seem restrictive at first, but can easily be extended to more complex camera configurations which lift the restriction on range of views. In configurations of more than two cameras, virtual views can be generated by taking the nearest two camera images as the base image set. In the next version of this project, we will expand the system to be able to handle eight cameras around the participant. Unfortunately, for this type of free motion configuration, it will not be possible anymore to use auto-stereo display and we will have to resort to passive stereo display. This implies that the burden of alignment in the face region must rely exclusively on the nose tracking or on some sort of targets located on the polarized glasses.

Table 1: *Comparison between features for view morphing*

Feature Used	Difference in RGB Values
Contour	9172
Contour and nose	3807
Contour and eyes	4878
Contour, nose, and eyes	2470

References

[1] E.H. Adelson and J. Bergen. ’The plenoptic function and the elements of early vision’. In Computational Models of Visual Processing, pages 3-20. MIT Press, Cambridge, MA. 1991.

[2] S. Avidan and A. Shashua. ’Novel view synthesis in tensor space’. In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pages 1034-1040, 1997.

[3] S. Baba, H. Saito, S. Vedula, K.M. Cheung, and T. Kanade. ’Appearance-Based Virtual-View Generation for Fly Through in a Real Dynamic Scene’, In VisSym ’00 (Joint Eurographics - IEEE TCVG Symposium on Visualization), May, 2000.

[4] H. Baker, N. Bhatti, D. Tanguay, I. Sobel, D. Gelb, M. Goss, J. MacCormick, B. Culbertson , T. Malzbender,’Computation and Performance Issues in Coliseum, an Immersive Videoconferencing System’. ACM Multimedia 2003, Berkeley, CA, November 2-8, 2003.

[5] S. E. Chen. ’QuickTime VR an image-based approach to virtual environment navigation’. Computer Graphics, 29(Annual Conference Series):29–38,1995.

[6] S.F. El-Hakim, J.-A. Beraldin, G. Godin, P. Boulanger. 'Two 3-D Sensors for Environment Modeling and Virtual Reality: Calibration and Multi-view registration'. International Archives of Photogrammetry and Remote Sensing. Volume 31, Part B5, Commission V. Vienna, Austria: 140-146. July 9-19, 1996.

[7] H. Fuchs, G. Bishop, K. Arthur, L. McMillan, R. Bajcsy, S. Lee, H. Farid, and T. Kanade. 'Virtual space teleconferencing using a sea of cameras'. Technical Report TR94-033, 18, 1994.

[8] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen. 'The lumigraph'. In Computer Graphics Proceedings, Annual Conference Series, pages 43-54, Proc. SIGGRAPH'96. August 1996.

[9] D. O. Gorodnichy. 'On Importance of Nose for Face Tracking'.Proc. Intern. Conf. on Automatic Face and Gesture Recognition (FG'2002), pp. 188-196, Washington DC, May 20-21, 2002.

[10] N. K. Hayles, 'How to Put Bodies Back in the Picture', Immersed in Technology: Art and Virtual Environments, Cambridge: MIT Press, 1995.

[11] H. C. Huangy. 'Disparity-based view morphing - a new technique for imagebased rendering'. ACM Symposium on Virtual Reality Software and Technology (VRST), Taipei, Taiwan, November 2-5, 1998.

[12] T. Kanade, H. Saito, and S. Vedula. 'The 3d room: Digitizing timevarying 3d events by synchronized multiple video streams'. Robotics Institute Technical Report, CMU-RI-TR-98-34. December 1998.

[13] S. Laveau and O. Faugeras. '3-d scene representation as a collection of images and fundamental matrices'. Rapport de recherche, Institut national de recherche en informatique et automatique, RR-2205, February 1994.

[14] M. Levoy and P. Hanrahan. 'Light field rendering'. In Computer Graphics Proceedings, Annual Conference Series, pages 31-42, Proc. SIGGRAPHŠ96. August 1996.

[15] J. Lanier. 'Virtually there'. Scientific American, pages 66Ű75. April 2001.

[16] B.J. Lei and E.A. Hendriks. 'Multi-step view synthesis with occlusion handling'. 2001.

[17] L. McMillan and G. Bishop. 'Plenoptic modelling: An image-based rendering system'. Computer Graphics, 29(Annual Conference Series):39Ű46, 1995.

[18] W. Matusik, C. Buehler, R. Raskar, S. Gortler, L. McMillan. 'Image-based Visual Hulls'. SIGGRAPH 2000, pp. 369-374.

[19] M. Pollefeys. 'Self-calibration and metric 3D reconstruction from uncalibrated image sequences', Ph.D. Thesis, ESAT-PSI, K.U.Leuven, 1999.

[20] S. Prince, T. Williamson, A. Cheok, F. Farbiz, M. Billinghurst, and H. Kato. '3-d live:real-time interaction for mixed reality'. In Proceedings of the ACM Conference on Computer Supported Collaborative Work (CSCW 2002), Nov. 16-20th , New Orleans, Louisiana. 2002.

[21] D. Scharstein. 'View synthesis using stereo vision'. PhD thesis, Cornell University, Technical Report CORNELLCS:TR96-1604. January 1997.

[22] S. Seitz and C. Dyer. 'Physically-Valid View Synthesis by Image Interpolation'. Proc. Workshop on Representation of Visual Scenes, IEEE Computer Society Press, June, 1995.

[23] S. Seitz and C. Dyer. 'View Morphing'. In SIGGRAPH 96 Conference Proceedings, Annual Conference Series, pages 21Ű30. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, August 1996.

[24] S.M. Seitz. 'Image-Based Transformation of Viewpoint and Scene Appearance, Ph.D. Dissertation', Computer Sciences Department Technical Report 1354, University of Wisconsin - Madison, October 1997.

[25] www.shapecapture.com

[26] G. Wolberg, H. Sueyllam, M. A. Ismail, and K. M. Ahmed. 'One dimensional resampling with inverse and forward mapping functions'. Journal of Graphics Tools, 5(3):11-33, 2000.

[27] C.R. Wren, A. Azarbaye Jani, T. Darrell, and A. Pentland. 'Pfinder: Real-Time Tracking of the Human Body', IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp. 780-785, July 1997.

[28] H. Y. Shum and S. B. Kang. 'A review of image-based rendering techniques'. IEEE/SPIE Visual Communications and Image Processing (VCIP) 2000, pp. 2-13, Perth. June 2000.