INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

University of Alberta

Design of a Prototype Tele–Immersive System Based on View–Morphing

by

Martha del Carmen Benitez Angulo

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the

requirements for the degree of *Master of Science*

Department of *Computing Science*

Edmonton, Alberta
Spring 2005

Dedication

To my father, mother, and sister for always being with me.
Thank you very much for your patience and support, I could not have made this without you. I love you.

ABSTRACT

Tele–immersive systems are a new generation of tele-communication technology aiming at giving to participants at a meeting (local and remote) the illusion that they are in the same room. Contrary to current videoconference technologies, tele–immersive systems will allow participants to a meeting to remotely express non-verbal communication clues: a critical element to create a real sense of remote presence.

To create a useful and realistic tele–immersive system, many problems have to be solved involving numerous areas of Computer Science and Engineering. This thesis describes the current state-of-the-art in this field and then defines a basic architecture for a prototype system based on an automated View–morphing algorithm. In addition to the automated view–morphing algorithm, one of the main contributions of this thesis is the systematic analysis of the visual clues necessary to solve the correspondence problem between two images.

Acknowledgements

I would specially like to thank Dr. Pierre Boulanger for his supervision of this work and for letting me be part of such an interesting project.

The economic support provided by Consejo Nacional de Ciencia y Tecnologia (CONACYT) is gratefully aknowledged.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recent developments in high-speed networking such as CanNet*4 [106], GEANT [36], and the Internet2 [121] to name a few, have provided new means of communication and new ways to exchange information at a speed and bandwidth that was thought impossible years ago.

In current applications, information is shared across different locations over band-limited networks by means of e-mail, simple file transfer protocol (FTP), or the Internet. Packets switching protocols such as TCP and UDP opened the door to a large number of applications taking advantage of the computer capabilities for communication.

Following this development, it became possible to share real–time text messages. Instant messaging applications became widely used with the introduction to the Internet of ICQ in 1996 [11] followed by many others [1, 4, 7]. Sending audio and video streams over the Internet became the next step, turning computers into a new communication device with new possibilities.

Videoconferencing systems integrated with desktop computers of fairly good quality are currently available on the market (Click to Meet [3], PictureTel [113], VCON [9], VTEL [10]), however they are not yet able to provide to their users the illusion of being really at a meeting, an effect called tele-immersion [64, 96] or tele-presence [82].

The question is: What are the basic factors necessary to achieve a life-like interaction and a sense of presence?

The notion of presence is truly a difficult concept to define and even more difficult to achieve. Many researchers in the field of VR [17, 122] and psychology [95, 101] study this notion and how to create this illusion. Fundamentally, the illusion of presence can be created by providing, by artificial means, a sense of being truly in a virtual environment, making him or her believe that it is real. This can be achieved by artificially creating an interface capable of creating a real-time sensory experience through the various human sensory channels. These sensory channels are: Vision, Audition, Touch, Smell, and Taste. There are many ways to create a sense of presence, but according to Burdea [26] they must include three basic elements: **interaction**, **immersion**, and **imagination**. By establishing a proper relationship between these so-called 3I's of VR, a sense of presence can be created.

The second question is: How can we translate this illusion of presence for a telecommunication device? How can we create the illusion that people located far away can meet in a virtual room and give them a "feeling" they are having a meeting in the same room? One of the solutions to this ambitious goal is tele-immersion. Let us define what tele-immersion means:

"Tele-immersion, a new medium for human interaction enabled by digital technologies, approximates the illusion that a user is in the same physical space as other people, even though the other participants might in fact be hundreds or thousands of miles away. It combines the display and interaction techniques of Virtual Reality with new vision technologies that transcend the traditional limitations of a camera. Rather than merely observing people and their immediate environment from one vantage point, tele-immersion stations convey them as "moving sculptures", without favouring a single point of view. The result is that all the participants, however distant, can share and explore a life-size space."[64]

Tele-immersion presents a series of challenges in many research areas in computing science such as computer vision, computer graphics and rendering, haptic systems, display technologies, tracking systems, and real-time network communications.

In order to achieve tele-immersion, each participant must have at least two viewpoints of remote sites to preserve depth perception, a key element for a sense of presence. Every participant and his/her surroundings must be sensed from all directions, this information is then sent to other participants and then rendered from their personal viewpoints.

Tele–immersive applications require significant computer resources for the acquisition, processing, and transmission of the information over high-speed networks. Currently this new and expensive technology is used for limited applications such as tele-medicine [18, 116], engineering design review [15, 94], and remote robot control [66]. Recent advances in computer and network capabilities point to a near future where a wide variety of applications ranging from casual conversation at the dinner table to virtual presence at a conference will be possible using this technology.

1.1 Proposed System

A tele–immersive system is a collaborative virtual reality application [23] that enables its users to share information across networks in real-time, providing them with the means to interact with each other as well as with computational models. Tele-immersion can be seen as a migration from Human–Computer–Interaction (HCI) to Human–Computer–Human collaboration (HCH).

There are many problems associated with the implementation of a tele–immersive system that need to be solved. In this thesis, we are addressing only a few of those problems, namely the problem of new view generation from a bank

of cameras. We will also describe the basic hardware and software architecture of a prototype system. In general a tele–immersive system is composed of the following elements:

- A system that senses the position of the participants in each location and establishes a common reference point where the participants can be located in a virtual meeting room.

- A network of video cameras to capture at least two different perspectives for each participant.

- A method that processes the video streams and produces a stereo pair of video according to each participant's position.

- A communication platform that allows the video and other multimedia information to be sent to other participants.

- A method that processes the information received and displays it according to participant's position with respect to the reference point.

Let us briefly describe how a tele–immersive system works in general terms and then more specifically describe a prototype system to test the concept.

In Figure 1.1, a simple sketch of the system is shown representing the communication between two users in different geographic locations. The system has the possibility to include in a meeting as many users as there are enough resources to process and transmit the information.

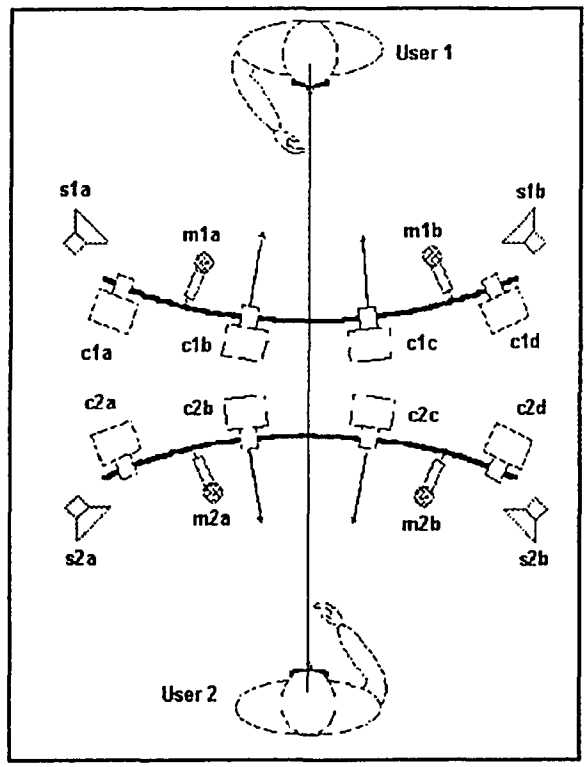


Figure 1.1: Proposed tele–immersive system [23].

As seen in the Figure 1.1, each user has a system of their own that is composed of a set of cameras (labelled c1x, c2x), microphones (labelled m1x and m2x) and speakers (labelled s1x and s2x). The simplest version of the system uses only two cameras to generate stereo images; however, if there are more cameras available in the system the number of virtual viewpoints that can be generated increases. In the case of the audio signal, at least one pair of microphones and one pair of speakers are used for audio communication between the users. Active echo cancelling [43] hardware is also used to improve sound quality.

A meeting starts when *User1* establishes a connection with *User2* through the network, thus establishing a session. At this point the system begins to acquire video and sound from each of the participants.

The system has a tracking module that records user position at all times during the session. This information is used to generate a stereo pair with adequate perspective corrections, from the receiver's viewpoint. In Figure 1.1, *User1* is facing *User2* thus the selected cameras are c1b, c1c and c2b, c2c, respectively. When the user moves in the real environment, the camera pair chosen will change, depending on which pair of cameras is closer to the user position provided by the tracker. Figure 1.2 shows this scenario when *User2* has moved to the right and the selected pairs are c1c, c1d for *User1* and c2c, c2d for *User2*.

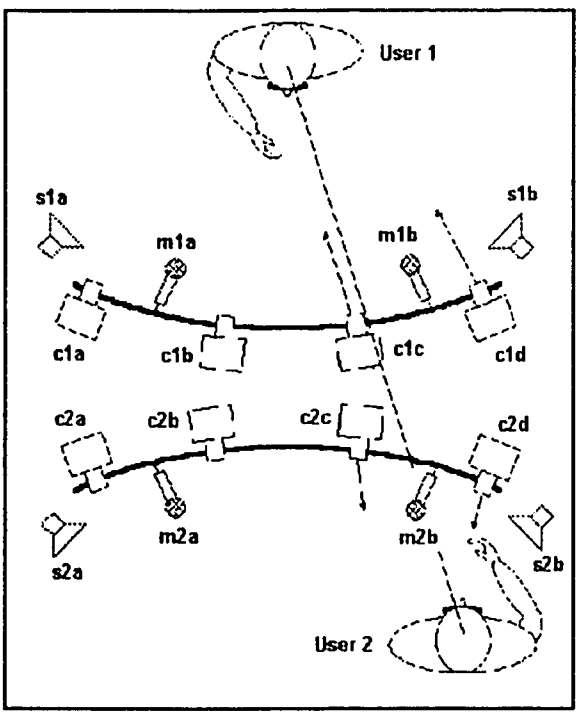


Figure 1.2: Selecting camera pairs for another viewpoint [23].

Video sequences from the selected camera pair, along with audio and user position, are compressed and sent over the network using compression schemes such as MPEG4 [63] or H.323 [110]. In the receiver, the video and audio signals are decompressed and processed for accurate presentation to the remote participants.

Processing starts by the identification of the participant's image in the video signal. This is done using a modified version of the PFinder algorithm [118] that performs a foreground/background segmentation.

Once the image of the participant has been extracted, a new stereo pair is generated both to adapt the separation of the images to the receiver's inter–ocular distance and to present the image of remote participants in the correct positions according to receiver's point–of–view. For the generation of new views, an implementation of a view–morphing algorithm [103] is used.

Position information provided by the position tracker is essential to system realism. It is this information that determines where to locate the rendered polygons representing the participant in the virtual meeting room (see Figure 1.3). Moreover, it allows the sound to be reproduced in the 3D virtual space reinforcing the sense of the other participant's locations.

Figure 1.3: Positioning of the video avatar in the virtual meeting room.

1.2 Prototype System

This thesis addresses the various modules found in the generic architecture of the proposed tele–immersive system shown in Figure 1.1. In this section, we will describe how an instance of this generic system was implemented using simple desktop display devices and a commercial hardware codec. One can see in Figure 1.4 a block diagram of the prototype system and in Figure 1.5 a picture of the system.

In this prototype, the system acquires images of the participants from two gen–locked cameras mounted on the auto–stereo display screen [2]. For the moment, the position of the participant is not measured because it is fixed due to the working principle of an auto–stereo display device, which requires the head position to be at a pre-determined distance called the "Sweet Spot." One can refer to [54] to get more details on the working principle of auto–stereo displays. The two images of the cameras are then combined using a side-by-side hardware multiplexer. The resulting stereo image pair is then combined with the audio signal and compressed using a hardware H.323 codec from Tandberg [6] and sent over a network at a rate of 1.5Mb/s.

At the receiving end, the signal is decoded and image processing on the stereo pair is performed to generate a new stereo pair from the receiver's point–of–view with corrected inter–ocular distance. The new stereo pair is then rendered on the auto–stereo screen with its corresponding sound.

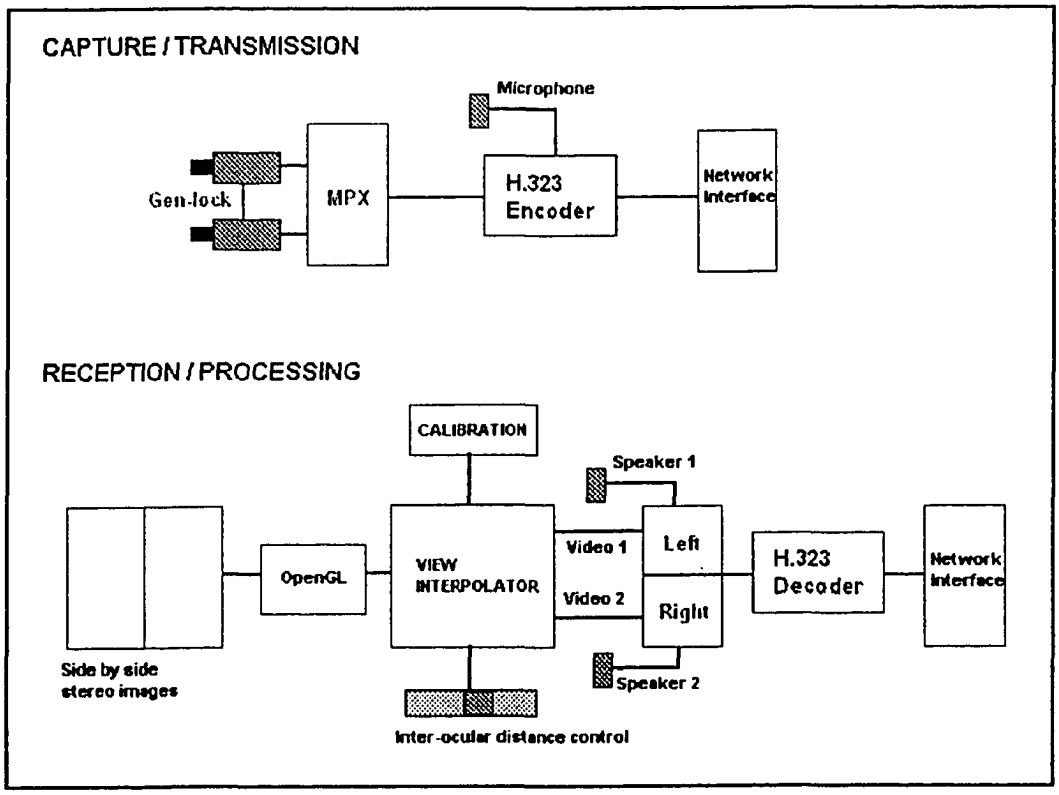


Figure 1.4: Prototype tele–immersive system block diagram.

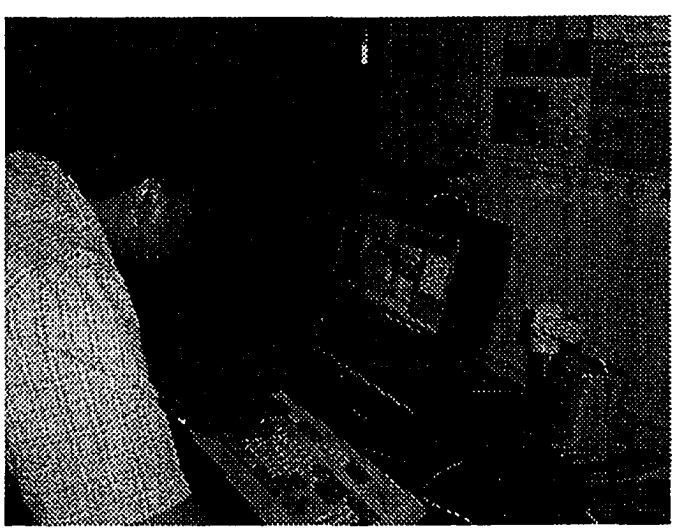


Figure 1.5: Picture of the prototype tele–immersive system.

1.3 Thesis Scope

One of the objectives of the current work is to review current trends in the development of tele–immersive systems and to analyze their main components.

This thesis gives special attention to the view generation problem, as it is key for the performance of the system. An implementation of view–morphing was developed for this work and integrated with the prototype system.

Results obtained from the view generation algorithm are analyzed to determine the quality of the results and the minimal set of features necessary to perform this difficult task.

It is beyond the scope of this thesis to discuss the implementation of all the modules of the proposed system. As mentioned previously, a hardware prototype of this tele–immersive system was built during the project with commercial equipment and was used for the experiments. However, the functions of many of the modules of the system will be simulated in order to focus on the performance of the view generation process.

1.4 Thesis Contributions

One of the contributions of this thesis is to review in detail, previous tele–immersive systems, discussing some of their pitfalls and describing how they can be overcome with the proposed system.

A significant part of the thesis deals with the view generation problem, a key element for a tele–immersive system. Depending on the positions of the users of the tele–immersive system, they will see different views of the virtual environment and the other participants. A different view of the scene also has to be generated every time a user changes his/her position or orientation.

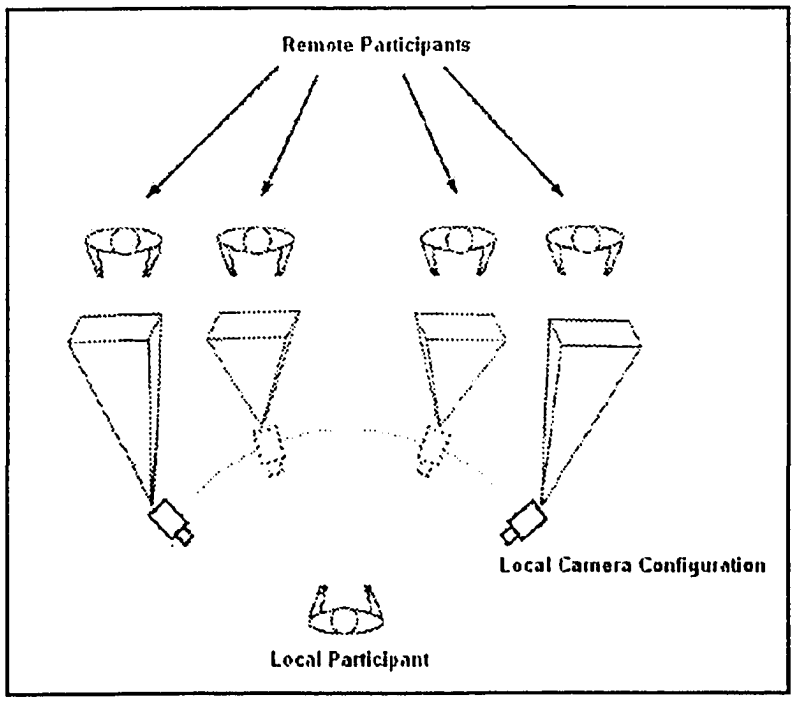


Figure 1.6: View generation problem.

We propose the use of view–morphing as the ideal solution to this problem for our system, and discuss how the limitations of the method are almost imperceptible for this application.

1.5 Thesis Outline

Chapter 2 sets the context of virtual environments in general. First, we explore different approaches that show how these environments are built and populated. Towards the end of the chapter some of the most representative tele–immersive systems found in the literature will be described. A brief discussion of their advantages and disadvantages is also included, as well as some aspects that could benefit our proposed implementation.

To better understand how the proposed tele–immersive system is going to work, each of its components is also analyzed in the following chapters.

In Chapter 3, we describe in detail the view generation sub-system, **our main contribution.** We demonstrate that View–morphing [103] is a very efficient alternative to solve, automatically and in real time, the view generation problem. All the details regarding the methodology are presented and whenever possible, related to the tele–immersive system under development.

Chapter 4 presents the acquisition system. Here the camera model and the calibration procedure are explained. We will also present how to solve the most common aberrations problem in imaging systems.

In Chapter 5, we present some methodologies for tracking objects in the image domain. Our attention is focused on background/foreground segmentation, a key element for our tele–immersive system. In this chapter we also describe some algorithms for facial features recognition. Particular emphasis is given to approaches for eye and mouth tracking, indispensable features to track in tele–immersive systems.

Finally, Chapter 6 presents a detailed description of the proposed tele–immersive system, mentioning all the implementation choices we made for each of the modules. In this chapter, we also offer some of the results obtained that confirm the direction of our work.

Chapter 2

Literature Review

Currently, with the development of new applications making use of virtual environments such as multimedia communication, 3D-telepresence, tele-robotics, etc., functionalities like continuous look–around and representation of 3–D data from different viewpoints have become indispensable.

Implementation of those more complex functionalities present new challenges in the development of new applications. The obvious way to solve the problem of sampling an environment is to use an array of cameras where each camera captures a different perspective of the environment. However, the problem is not trivial. Considerations like the cost of the devices, image storage and manipulation, and the possibility to process all the information in real time, impose limits on the amount of information that can be acquired.

Computer vision and computer graphics research in this area try to find new solutions to the problem making use of images to generate realistic renderings required for these applications, enabling the user to get new views from directions where no camera was placed during the sampling process.

Virtual environments (VEs), as their real world counterparts, are basically composed of two elements:

- World attributes: all objects in the scene.
- User attributes: graphic representation of the user's presence in the virtual world (avatar).

In the following sections, different approaches for the creation of these components of a virtual environment (VE) will be introduced and analyzed. From this analysis, we will be able to justify why view–morphing [103] was chosen in the present work as the basic technique for the implementation of a tele–immersive application.

This chapter also presents some outstanding tele–immersive systems in order to provide the reader with a context on the current state-of-the-art of this technology. We will also do a critical analysis of the differences between the system proposed in this thesis and the systems found in the literature.

2.1 Classification of Scene Representation Techniques

Virtual environments can be created and populated with objects using various techniques ranging from pure image–based methods to traditional 3–D graphic rendering methods. In the traditional computer graphics approach, 3–D models are rendered using various techniques (Phong Shading [88], Ray Tracing [13], Radiosity [45, 83], etc.) to produce an image of a 3–D scene. At the opposite end of the spectrum, computer vision attempts to reconstruct a 3–D model of a scene from a set of images [40]. In many research works, combining elements at both ends of the spectrum can yield a wide range of options to create photo-realistic VE in real time [67].

One can see in Figure 2.1, Milgram's [80] nomenclature of the continuum between pure image–based techniques and pure model–based techniques. Using this nomenclature, view generation techniques can be classified into three main categories: **model–based, image–based, and hybrid.**

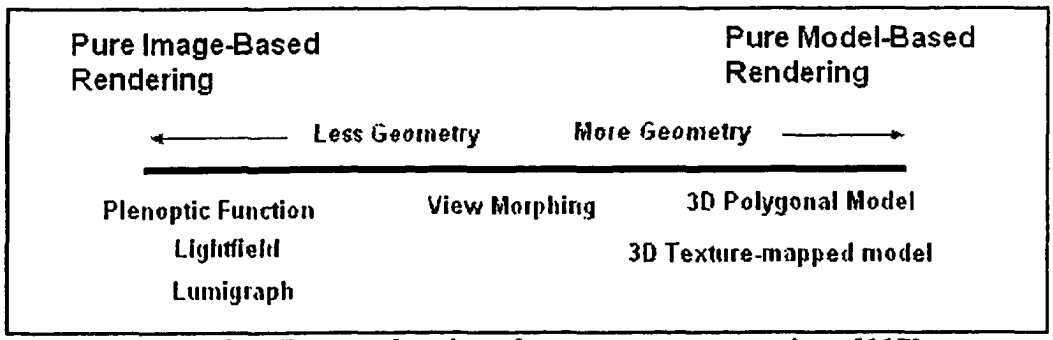


Figure 2.1: Range of options for scene representations [117].

It is important to mention that this classification is not the only one. There are some techniques that are difficult to classify using this nomenclature, for example: hybrid methods that concurrently use coarse 3–D models and image–based rendering, cannot be really classified as part of this continuum. Other classifications of view generation techniques along with their descriptions can be found in [67], [117], and [104].

2.1.1 Model–Based Representations

In this category, the first task is to construct a 3–D model of an environment. The model can be created both by hand using 3–D modeling software like Maya, 3D Studio MAX, and many others. It can also be reconstructed from real–world measurements using 3–D sensors. Depending of the resolution and the accuracy of the model one wants to create, various digitizing strategies can be used:

- Active scanning;
- Structure from stereo;
- Structure from motion;
- Structured lighting;
- Shape from shading.

Modeling real–world environments is a challenging task. All the objects in the environment have to be modeled separately with a high level of detail and then assembled in a large and complex scene graph. These modeling methods usually lack realism, and their complexity increases as the environment becomes larger or is populated by many objects. In many cases the resulting model becomes prohibitively large and computationally expensive to render.

In addition, when the model is obtained from 3–D sensor data, there are additional problems associated with this process. For instance, the processes of view registration, view integration, and model construction are difficult to implement and require large modeling software packages such as Polyworks [105]. Many of these techniques are sensitive to sensor noise and its physical limitations. The reader can refer to [62] for more details on the various processes associated with sensor based modeling.

Once the basic 3–D structure is built, pixel values from images can be used to texture the 3–D model in order to make it more photo-realistic. Using this modeling technique, a new view from a specific viewpoint can be generated by simply applying a rigid transformation to the scene graph that is then rendered.

In [58] a wireframe model of an object is built from the edge information extracted from a set of sample images. Disparity estimation is used to assign depth values to the model. Once the 3–D model is built, it can be rotated to any orientation and its surfaces texture–mapped with information from the images. At the end, a rendering algorithm can generate at any arbitrary viewpoint, a new image.

One of the most important works in this category, is view–dependent texture mapping. Devebec *et al.*, [38] applied this technique to the reconstruction of architectural environments.

In their work they use a photogrammetric modeling method to recover the basic 3–D geometry of an architectural scene. Then, a model–based stereo algorithm is used to recover accurate disparities from widely spaced images. Finally for the rendering phase, view-dependent texture maps are used. By combining multiple textures from different sampled viewpoints, geometric details can be improved on the models. This algorithm creates textures that are mapped onto the model surface by making an average of previous textures using weights based on their distance.

Another well–known project is the 3D Room developed at Carnegie Mellon University [60, 97]. This room is equipped with forty–nine cameras to capture events in real time, inside the room. A cluster of PCs digitizing all video signals from the cameras is used to record simultaneously the various video streams.

One of the main uses of this system is in the construction of realistic avatars [33]. Using the recorded video sequences from the cameras, a voxel model of a person is built using a shape from silhouette technique. The model is then converted into a smooth surface using a modified version of a marching cubes algorithm. Due to the complexity of processing a large number of video streams to create the 3–D model, it is currently impossible to use this system for real time applications.

The main disadvantage of model–based techniques is that in most cases it is very difficult to obtain a complete 3–D structure of a scene in general environments. The best results are usually obtained in office environments or with scenes with low complexity. In addition, because of the complexity of the calculation, real-time versions are currently prohibitive in cost and computational power.

2.1.2 Image–Based Representations

The second alternative is image–based rendering, where a new view is generated directly from multiple images representing the scene. In this category, no 3–D reconstruction is performed, so the rendering does not depend on scene complexity but only on image resolution and the number of cameras used to digitize the various viewpoints.

For these techniques, the problem of view synthesis is not as trivial as it is for model–based approaches. As stated previously, new views for model–based techniques are simply generated by a rigid transformation followed by a standard computer graphics rendering process. For image–based representations, the novel view has to be generated from the discrete views.

Pure image–based approaches are based on the use of the plenoptic function [12]. Let us name P the function that describes a scene in terms of seven basic parameters: 3–D location (V_x, V_y, V_z), orientation angles (θ, ϕ), light wavelength λ and time t. The intensity of the light rays at position p is given by:

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

The plenoptic function represents all possible views of a scene at a particular position in space–time for a specific wavelength. In order to sample this function, an unrealistic amount of image data would need to be captured in order for any viewpoints to be generated.

Given the fact that it is impossible to obtain a complete plenoptic function with current technologies, different alternatives have been proposed for image–based view generation from a simplified version of the plenoptic function.

McMillan and Bishop [78] introduced the plenoptic modeling, a five-dimensional plenoptic function that assumes a static environment, thus wavelength λ and time t are constant .

Simplifying the plenoptic function a step further, light fields [69] and the lumigraph [48] were introduced almost simultaneously. By limiting the possible viewpoints to a bounding box, a scene can be represented in terms of four parameters. Images in the light fields method are obtained by applying some filtering and interpolation to move between the input images, while in a lumigraph, some geometric knowledge is used to compensate for non–uniform sampling of the image set.

The simplest plenoptic function can be represented by a 2–D panorama, either cylindrical or spherical, where the viewpoint is fixed and the only parameters to model are the angles (θ, ϕ). Based on this function Apple developed Quick Time VR, a commercial application that renders static environments [32].

Table 2.1 shows the taxonomy of plenoptic functions as proposed by Shum [104]. The main advantages of image–based representations include the realism obtained from the direct use of images and the fact that the cost of generating a new view does not depend on scene complexity. However, the big drawback of this category of algorithms is the difficulty of acquisition and storage of large image sets.

Name	Dimensionality	View Space	Year
Plenoptic function	7	Free	1991
Plenoptic modeling	5	Free	1995
Light Field / Lumigraph	4	Bounding box	1996
Concentric Mosaics	3	Bounding circle	1999
Cylindrical / Spherical Panorama	2	Fixed point	1994

Table 2-1: Different plenoptic functions for image–based methods.

In particular, for the kind of system we are proposing in this work, real-time scene representation is crucial. Currently, image–based techniques only work for static environments.

2.1.3 Hybrid Representations

The last category in this classification, is represented by hybrid methods (or transfer methods as named by the photogrammetric community). These methods use both geometry and image information to generate new views. This category is also known as rendering with implicit geometry [104].

View generation is done by some processing of positional correspondences from a small image set and some geometric constraints used to re-project image pixels at a given virtual camera viewpoint. The geometric constraints, either known *a priori* or computed in an early stage, usually take the form of depth values at each pixel computed by epipolar constraints or trilinear tensors.

Chen and Williams [31] were among the first ones to synthesize views from a set of images with their View Interpolation technique. Using range data to establish the correspondences among images, they interpolate offset vectors to generate novel views.

Another approach is view–morphing, [103] technique that is able to reconstruct any intermediate view from images along the line that joins the camera centres. The importance of this technique lies in the fact that contrary to view interpolation, it ensures that intermediate views are a mathematically correct combination of the input images.

Other research works use a modified version of view–morphing methods. Park and Yoon [87] add the use of sprites and claim that this augmentation solves the occlusion problem. Bao and Xu [16] make use of the fact that wavelets localize features both in the image and frequency domains. They add a fourth step to view–morphing to transform the images into wavelets and then interpolate the wavelet coefficients. The drawback of this method is that resolution decreases with the change between image and wavelet domains. Scharstein *et al.* [99] extend the view–morphing framework to generate views along the plane joining camera centres.

Some other works relevant to our tele–immersive goals are view–morphing methods for dynamic scenes and video sequences. Manning and Dyer [73] showed that view–morphing could be used in dynamic context interpolating view and scene motion. Radke *et al.* [91] apply view–morphing methods to interpolate anchor frames for low bit-rate video applications.

Another common method, and one of the first that was used with natural scenes, is the work by Laveau and Faugeras [65]. They made use of the fundamental matrix to find a warping function to produce perspective correct views from the input images acquired by uncalibrated cameras. Five

corresponding points were specified by the user to determine the fundamental matrix allowing for the re-projection of pixels between two images.

Avidan and Shashua [14] use the trilinear tensor instead to establish point correspondences between three images for view generation. One of the advantages of this method is that if a trilinear tensor is known for a set of three images, given a pair of point correspondences, the third corresponding point can be calculated directly. In addition, the trilinear tensor is more stable than the fundamental matrix used by Faugeras and the specification of a new view is more direct.

Within the hybrid method categories, there is an endless list of methods, each of them trying to solve a particular problem. One possible classification of the hybrid methods, based on some of the main differences between them, is shown in Table 2-2.

▪ Work in real–time	Using special hardware [75]
▪ Need offline processing	[56], [30]
▪ Methods to obtain depth information or disparity:	
• Block matching	[58, 85, 86, 123]
• Correlation-based / feature-based	[87]
• Predictive and multistage	[71]
• Template matching	[56]
• Hierarchical	[107]
▪ Type of interpolation:	
• Linear	[56]
• Nearest-neighbour / linear	[84]
• Winner-takes-all	[74]

▪ Interpolate images in time	[91], [73]
▪ Interpolate images in space	(most of the techniques)
▪ Work with two or multiple images	
▪ Methods to deal with occlusions:	
• Sprites	[87]
• Reconstruction from furthest object	[123]
• Joint view triangulation	[70]
▪ Methods to solve correspondence:	
• Definition of a 2–D space	[75]
• Quasi–dense matching	[70]
• Aspect graphs	[55]

Table 2-2: Different characteristics of hybrid methods.

2.2 View Generation for Tele-Immersion

Considering the context of this thesis, hybrid methods seem to be the best option for view generation given the fact that they avoid the problems of model–based approaches (performance, lack of realism and dependence on scene complexity) and those of image–based methods requiring complex data acquisition systems, large storage capabilities, and its current inability to work in real-time.

Having in mind the advantages of hybrid techniques, the realism of the results obtained by Seitz and Dyer [103] and the simplicity in the implementation of these algorithms, we chose view–morphing to generate the views for the system discussed in this thesis. A detailed description of view–morphing and the issues associated with its implementation will be discussed in Chapter 3.

2.3 *Avatar Representations*

In the context of virtual reality, an avatar is a graphic representation of human form. Avatars can be considered a special case of scene representation and a large body of research exists to study how to convey the embodiment of a user in a virtual world.

An avatar must meet different requirements depending on the application for which it is being used. In some cases a simple cartoon-like avatar may be enough to represent a user while in other cases the actual representation of true facial expressions of the user may be needed. In general, the best avatars are those that represent the user realistically, giving him/her the feeling of being immersed in a virtual environment.

In particular, for communication systems, the tendency in avatar development is to create a natural representation of the participants with the ability to communicate the subtle elements present in a real–world interaction such as: facial expressions, lip movement, body postures, and gestures.

Representations of humans in virtual environments determine the way users perceive each other in the virtual environment, and are a key factor in system quality. The first systems had very primitive representations of the users ranging from cube-like appearances [50] to rigid cartoon-like avatars [22]. These computer–generated models evolved into articulated bodies [19, 29, 90] and finally fully deformable animations capable of representing facial and gesture communication [28]. Various avatar technologies are illustrated in Figure 2.2.

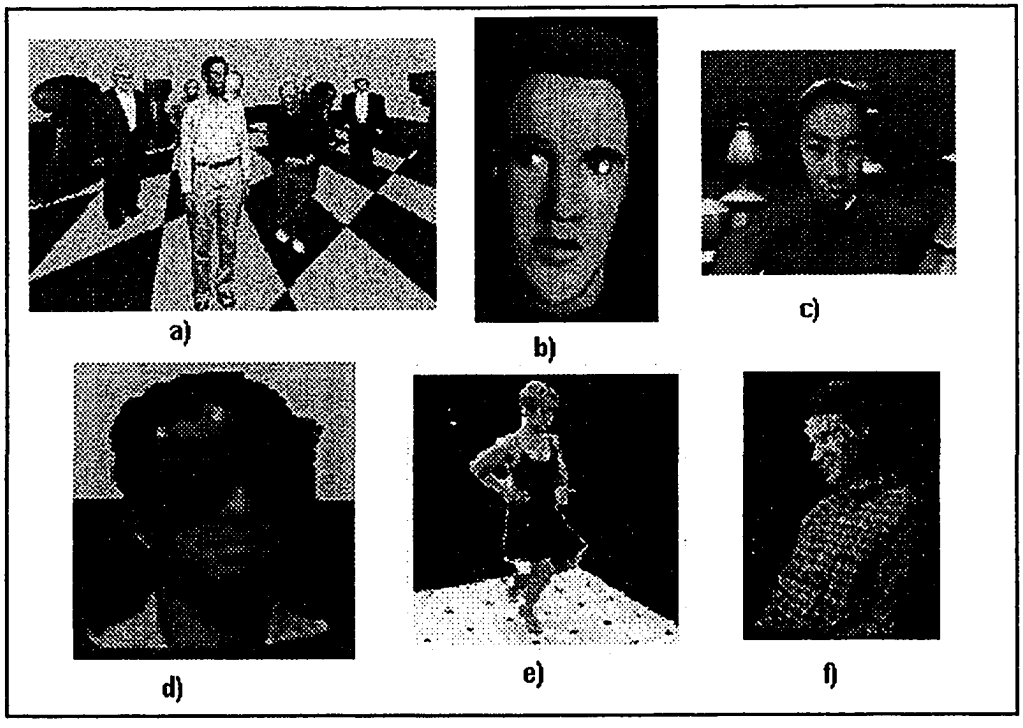


Figure 2.2: Various avatar representations: (a) video avatar [57], (b) computer–generated avatar with facial expressions, (c) with texture–mapped video [28], (d) texture–mapped pre-scanned head model [92], (e) 3–D reconstructed dancer [33], and (f) 3–D reconstructed avatar [82].

In order to increase the realism of avatars, the idea of inserting the video of the user in the virtual environment emerged [44, 57]. In its simplest form, a single camera is used and video is texture–mapped onto a polygon. Although this provides the image of the user and his/her gestures, all the realism of 3–D clues is lost, thus losing any sense of immersion or presence. To improve video avatars Boulanger *et al.* [24] suggest the notion of stereo texture where two real-time video textures are alternated on a polygon during rendering time. This stereo texture works quite well for normal views but does not allow navigation around the avatars.

Going a step further to provide depth perception, some other techniques [68, 98, 120] map video onto generic 3–D models. In this category, we find the

work by [28] where a computer–generated 3–D model of a person is created and then, given the localization of the features on the face, the video from the user is texture–mapped onto the model. There is also the possibility to create the actual model of the user from the video cameras [92]. In this case, images of the user are acquired while he is rotating in front of a camera at intervals of approximately 45 degrees. These images are then used to build a coarse 3–D model using a volumetric intersection based on silhouettes. The video of the user is then texture–mapped onto the head model, to represent the facial expressions of the user. For these techniques, precise feature alignment is needed in order to obtain a good representation of the user.

Some other techniques avoid the use of computer-generated models and use a set of images of the user to build the 3–D model. One example is the work [33] performed at Kanade's Virtualized Reality Laboratory at CMU where the avatar of a dancer is created from video sequences acquired from forty–nine different cameras. As discussed previously, the avatars produced from this model–based technique are not real-time and require very expensive calculations. In this category of method, one should mention the work by [82] that acquires the 3–D model of a person from a rig of seven cameras in real-time.

The main issues with avatar representations are that computer–generated models are not able to represent all the details of a user and convey their behaviour. As suggested previously, sensor–based methods must be used to improve the realism. However, to obtain the best models from video sequences, a large number of cameras is needed or some processing offline has to be done, making the use of these systems difficult, if not impossible to work in real-time.

One possible solution to the real-time problem is to make use of a view synthesis technique to create the avatars. Using a technique like [103] it is possible to render a stereo pair of the user and insert these images in the virtual environment.

2.4 Tele-Immersive Systems

As mentioned previously, the idea of tele-immersion implies an improvement over current videoconferencing systems to improve the sense of presence.

Current videoconferencing systems work by setting a video camera and monitor at each participating site, by capturing a single perspective of the environment and then sending the video stream and audio signal through the network for the other participants to see. The video image is then displayed on the computer screen of each receiving site.

Those systems have many problems, one of the key problems is that they are totally incapable of creating a sense of tele-presence. First, the user has to adapt his behaviour in order to fit in the video window creating significant restrictions on his/her ability to interact with other participants or with data. Another difficulty of videoconference is that each participant is represented by multiple video windows, thus further reducing the sense of presence. One can see in Figure 2.3 a screen shot of a recent videoconference at the UofA Access Grid [8, 112] node.

Figure 2.3: Typical desktop view of an Access Grid meeting.

However, the disadvantages discussed previously are not the most damaging for communication because the user adapts to these limitations easily. The biggest problem lies in the loss of human interaction. While video allows each user to see the other participants, it is impossible to achieve eye contact due to the fact that the camera and the display cannot be at the same place. This result in a loss of focus–of–attention to the display toward his/her real interlocutor: the camera.

Moreover, in everyday interaction, people do not get all the information just from verbal information. Lots of the information received in a conversation is in the body language and physical interaction between people.

2.4.1 First Tele–Immersive Systems

Recently, research projects like [61] began to include immersive video with the idea of creating a 3–D model of an environment in which a viewer can move around. A set of images is acquired from real cameras and the walk–through is rendered by computer–generated images of intermediate locations.

In this context of generating virtual environments from the processing of multiple video perspectives there is also the system developed by Moezzi *et al.* [81]. The system patented by this group builds a 3–D model of the environment from multiple scene images while also detecting and tracking objects and their locations. The main disadvantage of this approach, as well as for all the implementations that work by building 3–D models, is that it is computationally intensive, slow, expensive, and do not scale well with scene complexity.

Another approach in the development of virtual meeting applications is one that tries to synthesize an environment from both real and computer–generated elements. In this category, we can mention the work performed by McNerney *et al.* [79]. This system renders an emulated conference room in which each participant is represented by his/her own image located in a specific chair. While this attempt to achieve realism gives the participants some feeling of being in a meeting, it also limits them because they are unable to move around the room. Moreover, the fact that people are represented by an image, gives the participants the feeling of being outside of the meeting.

2.4.2 National Tele-Immersion Initiative

While there are many research groups working in the development of techniques to achieve the feeling of immersion in a virtual meeting, the most important and representative work so far has been done by the National Tele-Immersion Initiative (NTTI [96]).

The NTTI is a research group in the United States with collaborators from Advanced Network Systems, the University of North Carolina at Chapel Hill, the University of Pennsylvania, and Brown University. Their goal is to establish a standard technology for the development of tele–immersive applications.

Work by the NTTI is centred in the development and improvement of a seven camera system for the creation of realistic avatars. They make use of structured light, imperceptible to the human eye, that is captured by the synchronized cameras and then used for the reconstruction of the 3–D avatar. Currently the updates rate for the avatar reconstruction is two to three times per second.

Figure 2.4: NTTI seven camera system (left) and 3–D avatar (right) [82].

Henri Fuchs and his team at Chapel Hill, have presented the most clear perspective of what should be achieved by a tele–immersive system, and describe all the ideas in detail, as well as their results to date [93]. An image of the current implementation is shown below.

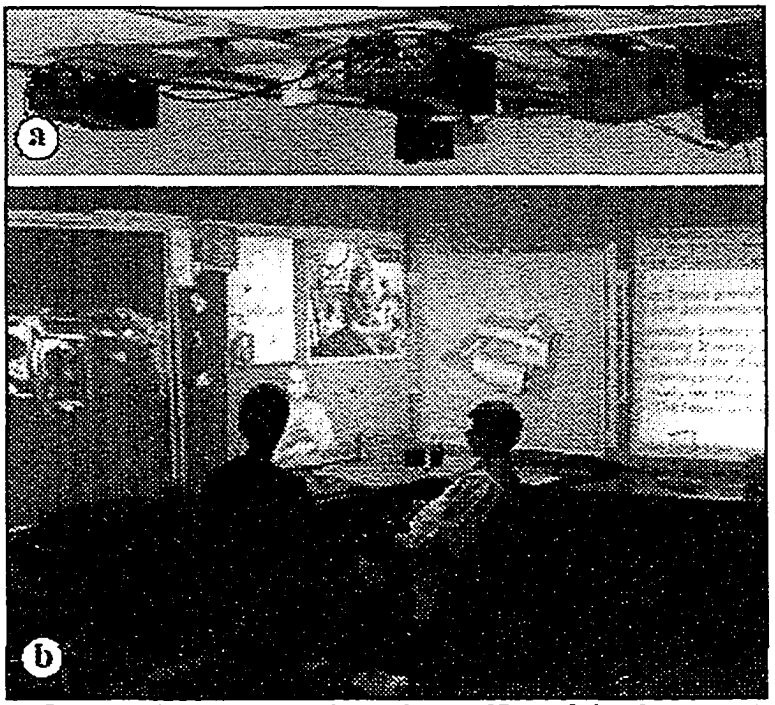


Figure 2.5: Current implementation of the office of the future: a) cameras and digital light projectors, b) walls used as immersive displays [93].

The office of the future, as described in their work, is a complete framework that makes use of computer vision techniques, computer graphics, and image–based modeling to provide a traditional office with the potential to become a virtual meeting room.

The main components of the office of the future are a set of video cameras to capture the office and people in it, and a set of intelligent projectors used both to control the light in the office and to render video from remote locations. This configuration allows simultaneous capture and display of video sequences. Images can be projected onto surfaces, such as a wall, that become a window into the virtual meeting room.

2.4.3 The TelePort Project

Another outstanding system aiming to achieve tele-presence is the work developed by Gibbs *et al.* [44].

The main idea behind the TelePort project is to simulate a real–life meeting where the users gather in the same room to interact. To make this possible, they combine the real world with a virtual one that is an extension of the former.

A special room is configured for the videoconference. In this system one of the walls is used as the rendering surface in order to have a life size display. On this wall the virtual extension of the room is rendered with the remote participants inserted in it.

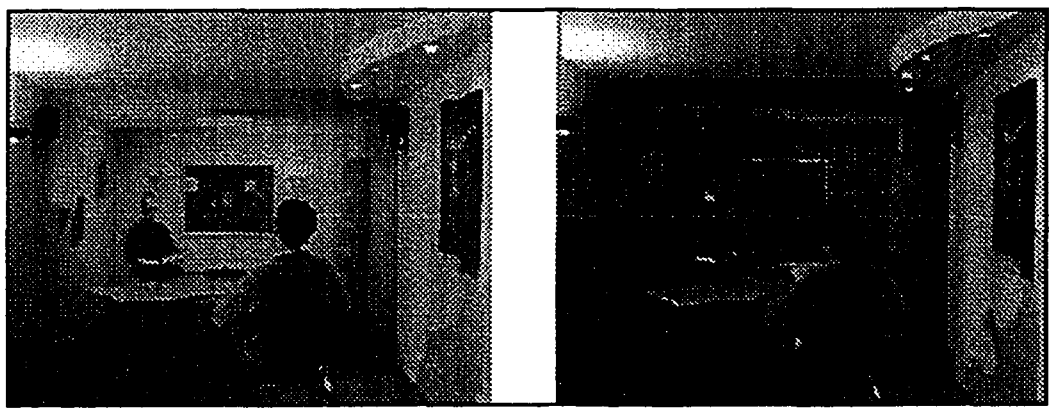


Figure 2.6: A TelePort session: virtual extension as perceived by local user (left) and virtual extension rendered in wireframe (right) [44].

The system uses stereo video sequences to acquire the user representation. Video is processed to segment the user from the background and the resulting image is texture–mapped on a polygon that is inserted in the virtual environment. Rendering is performed according to the user's position information provided by a tracking system in the room.

2.4.4 The Microsoft i2i Project

Microsoft's research group at Cambridge has also developed a system to enhance desktop videoconference applications. Their system, named i2i, consists of a pair of synchronized web cams connected to a common PC.

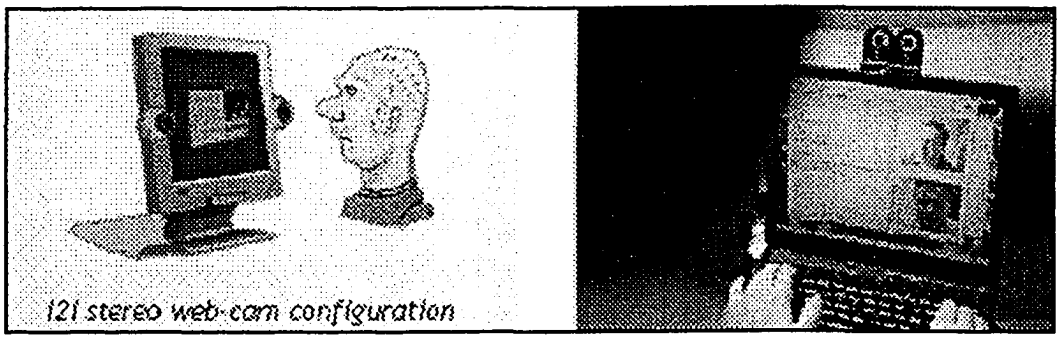


Figure 2.7 The i2i desktop system [34].

The various real–time vision algorithms this group is working on [35], make it possible for new functionalities to be included in a normal videoconferencing system:

- Eye-gaze correction: allowing the users to improve their interaction with each other;
- Background substitution;
- 3–D object insertion;
- High quality view generation;
- Automatic camera management.

The central process of their application is the view generation algorithm, with emphasis on recovering the continuous object boundaries without artifacts along them.

2.4.5 The HP Coliseum Project

The work done by HP Laboratories in Palo Alto [51, 52] is the most similar to the system we are proposing. The Coliseum project aims at creating a desktop immersive videoconferencing system that can provide realism in communication.

Coliseum has been designed and tested for multiple users and results obtained so far show that it is a step closer to the experience of a face-to-face meeting. The main idea behind its design is to generate a virtual environment where users will be inserted, with the ability to interact between them and even move within this environment.

The system consists of a rig of five video cameras attached to the LCD monitor of the PC where the user is working to provide video and a microphone and speakers for the audio.

To obtain the different representations of the user that will be rendered on each of his/her interlocutors' computers, a 3–D model of the user is built. Video images from the five cameras are processed first to segment the foreground (the user). Silhouette contours are then used as input for the image–based visual hulls technique that constructs a shape representation that will be used to render from the desired viewpoints at remote sites.

VRML is used to generate and render the virtual environment. Within the environment, users can change the relative position to each other that allows them to move around, and with head–tracking information, eye contact possible.

Figure 2.8: Coliseum immersive system [51].

2.5 Discussion

In the previous sections, many tele–immersive systems were presented. Although technologies are getting closer to provide users with alternatives to real–life meetings, there is still a lot of work to do in this area, both in the development of better and faster algorithms for image processing, as well as in the research of new hardware alternatives that enhance the experience.

Our system differs from the NTTI and TelePort in the fact that ours is a desktop system, whereas the other two are thought of as room-sized systems. One of the big challenges faced by a system of this size is the difficulty of sensing the environment and the participants from enough viewpoints and to process all that information in real–time. We believe that real-time View–morphing may change this situation.

In the following chapter, we will describe in detail how view–morphing works and how the basic algorithm can be modified for automation.

Chapter 3

The View–Morphing Algorithm

Different approaches looking to solve the view synthesis problem were discussed in the previous chapter, considering their main characteristics as well as their suitability for virtual environments.

In this chapter a view–morphing technique developed by Seitz and Dyer [103] is described in detail, mentioning the issues that limit its scope as well as its use in the context of the present work.

3.1 Morphing for View Generation

As mentioned previously, a robust technique for the generation of new views from a pair of images is essential for a tele–immersive system. This is exactly what view–morphing does, surpassing other techniques because of its simplicity of implementation and its ability to extend the method to fulfill particular needs.

The development of view–morphing started with the realization that morphing techniques were able to generate surprisingly life–like intermediate images and, in some cases, were able to convey the idea that objects in the scene had undergone 3–D transformations.

In general, morphing techniques did not guarantee that the resulting images preserved the shape of the objects. The reason for this is that linear interpolation does not preserve the projective mapping, thus the information relating the two views is distorted through the morphing process (usually bending straight lines).

In order to apply a morphing transformation to a pair of images for view generation, one must make sure that the intermediate views keep the same structure as the input images.

Let us start by describing how a morphing operation is defined. First we need a pair of images, I_0 and I_1, and two maps $C_{01} : I_0 \Rightarrow I_1$ and $C_{10} : I_1 \Rightarrow I_0$ establishing the correspondences between the two images. Two maps are used because the correspondences may not be one–to–one.

Depending on the morphing technique used [20, 114], the correspondences between the two images are specified in different ways by lines, meshes, or points usually selected by user interaction on the input images. Once a sparse set of correspondences is determined, the correspondences of the remaining pixels are obtained by interpolation.

From the correspondence maps, a warp function for each image is computed in order to displace the points to their desired position in image I_s. The warp functions based on linear interpolation are:

$$C_{0s}(p_0) = (1-s)p_0 + sC_0(p_0) \qquad (3.1)$$

$$C_{1s}(p_1) = (1-s)C_1(p_1) + sp_1 \qquad (3.2)$$

where C_{0s} and C_{1s} indicate the displacements of each point $p_0 \in I_0$ and $p_1 \in I_1$ in terms of the interpolation parameter $s \in [0,1]$. Each intermediate image I_s is generated by averaging the colours of the warped images C_{0s} and C_{1s}.

3.2 Shape-Preserving Morphs

In order to use morphing to emulate correct 3-D transformations, the input images must meet some conditions. In general, most image pairs can be classified into two main categories arising from the camera positions' geometry.

3.2.1 Parallel Views

Chen and Williams [31] were the first to study this particular case. Here the camera moves parallel to the image plane. They argued that this kind of motion produces new perspective views.

In detail, this scenario can be described as follows. Consider a camera located at position C_0 acquiring the image I_0 of an object, then the camera is moved in a direction parallel to the image plane to position C_1 while zooming out to obtain the second image I_1, as shown in Figure 3.1.

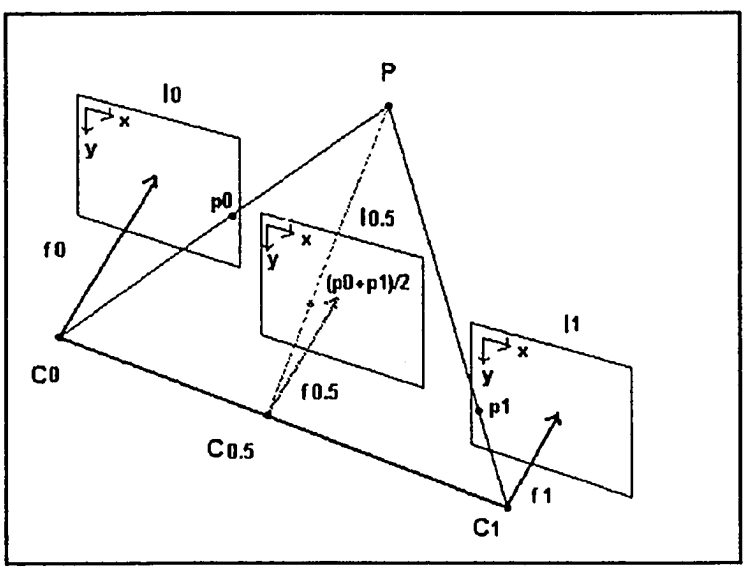


Figure 3.1: View-morphing: parallel views [103].

From the image in Figure 3.1, it is easy to describe this scenario with the corresponding mathematical equations representing the image planes. Assuming that the camera's position C_0 corresponds to the world origin and it is moved to position $C_1 = (C_X, C_Y, 0)$, and the focal length was modified from f_0 to f_1. The projection matrices of the input images are given by:

$$\Pi_0 = \begin{bmatrix} f_0 & 0 & 0 & 0 \\ 0 & f_0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (3.3)$$

$$\Pi_1 = \begin{bmatrix} f_1 & 0 & 0 & -f_1 C_X \\ 0 & f_1 & 0 & -f_1 C_Y \\ 0 & 0 & 1 & 0 \end{bmatrix} . \qquad (3.4)$$

Any configuration described by similar projection matrices gives origin to *parallel views*, a special case where linear interpolation can be used to generate intermediate views of a scene without deformations.

For any 3–D point $P = [X, Y, Z, 1]^T$ represented by $p_0 \in I_0$ and $p_1 \in I_1$ in the input images, the interpolation produced by a morph operation is expressed by:

$$\begin{aligned} (1-s)p_0 + sp_1 &= (1-s)\frac{1}{Z}\Pi_0 P + s\frac{1}{Z}\Pi_1 P \\ &= \frac{1}{Z}\Pi_s P \end{aligned} \qquad (3.5)$$

where:

$$\Pi_s = (1-s)\Pi_0 + s\Pi_1 \qquad (3.6)$$

$$= \begin{bmatrix} f_s & 0 & 0 & -f_s\alpha_s C_X \\ 0 & f_s & 0 & -f_s\alpha_s C_Y \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.7}$$

$$f_s = (1-s)f_0 + sf_1 \tag{3.8}$$

$$\alpha_s = \frac{sf_1}{(1-s)f_0 + sf_1} \tag{3.9}$$

As can be seen from the previous equations, the new projection matrix Π_s is a linear interpolation of the projection matrices of the input images. It represents a camera with focal length f_s and projection centre $C_s = (\alpha_s C_X, \alpha_s C_Y, 0)$.

All images obtained from this parallel configuration are guaranteed to be a geometrically correct representation of the input views due to the fact that they represent views that would be obtained by a real camera while traversing the line C_0C_1 and zooming continuously.

In this context, the only thing that needs to be considered when selecting the input images is the visibility of all the objects in the scene in both images.

3.2.2 Non-Parallel Views

The previous section described how morphing techniques can be used to generate intermediate views from parallel cameras conveying a change in viewpoint without the use of 3–D information.

However, in real life and thus for real applications, not all the images are parallel. Here, an important property of projective geometry will can be used: the fact that any two views that share their optical centre are related by a planar projective transformation.

Based on this property of the images, *re-projection* can be used to modify the orientation of the image plane (gaze direction) of each input image in order to bring them into the parallel view configuration.

3.2.2.1 Image Re-projection

Consider two images I and $\hat{I}$ with projection matrices $\Pi = [H \mid -HC]$ and $\hat{\Pi} = [\hat{H} \mid -\hat{H}C]$ respectively. Any scene point P is projected to $p \in I$ and $\tilde{p} \in \hat{I}$. Then, the relationship between the views can be expressed by:

$$\begin{aligned} &\hat{H}H^{-1}p = \hat{H}H^{-1}\Pi P \\ &= \hat{\Pi}P \\ &= \tilde{p} \end{aligned} \qquad . \qquad (3.10)$$

In Equation 3.10, the 3x3 matrix $\hat{H}H^{-1}$ represents a projective transformation that re-projects the image plane of I into $\hat{I}$.

To make the most efficient use of the re-projection operation, it is necessary to choose an adequate coordinate system to reduce image distortion. A good option is to set both C_0 and C_1 on the X axis and select the Y axis in the direction of the cross products of the normal of the images. Once the input images are rectified, they can be morphed assuming the parallel view case.

3.2.2.2 A Three-Step Algorithm

The view–morphing algorithm for non-parallel views consists of three steps:

1. Prewarp: find transformations H_0^{-1} and H_1^{-1} and apply them to the input images to bring image planes into parallel configuration;
2. Morph: generate image $\hat{I}_s$ by linearly interpolating the positions and colours of the warped images;
3. Postwarp: apply transformation H_s to $\hat{I}_s$ to obtain the final image with the desired orientation.

This simple algorithm is represented in Figure 3.2 where the different transformations of the image planes are represented. First images I_0 and I_1 are brought into the parallel planes $\hat{I}_0$ and $\hat{I}_1$. The parallel views are then morphed to generate the $\hat{I}_s$ that is finally warped to the desired view I_s.

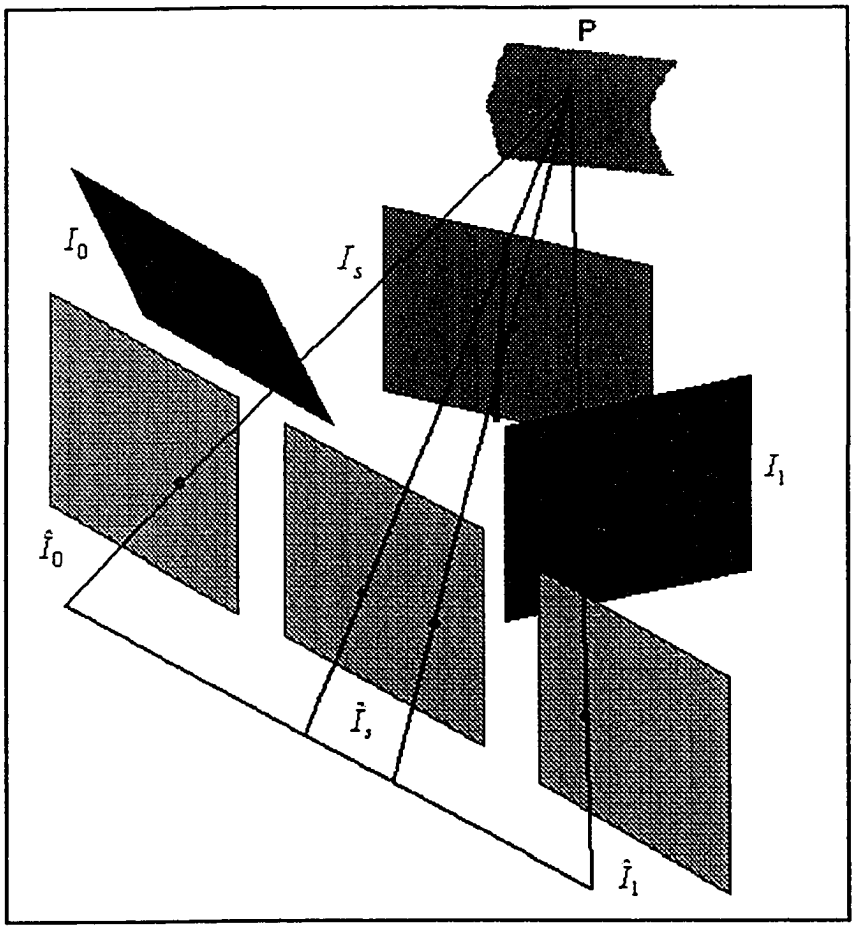


Figure 3.2: View–morphing: non-parallel views [24].

A great advantage of the three-steps algorithm outlined previously is given by the special form of the projection matrices of the rectified images $\hat{\Pi}_0 = [I \mid -C_0]$ and $\hat{\Pi}_1 = [I \mid -C_1]$. This configuration makes corresponding points appear in the same scan line in both images, thus simplifying the interpolation problem by performing it one scan line at a time.

3.2.2.3 The Case of Singular Views

There is a special case for non-parallel views that must be analyzed: the case of singular views.

This configuration arises when the motion of the camera is close to being parallel to the viewing direction. In this case, the centre of projection of one of the cameras appear in the field–of–view of the other, making it impossible to make the images parallel using rectification. For images that fall into this category all operations: prewarp, morph, and postwarp are combined into a single warp, to avoid explicit construction of prewarped images.

3.3 Critical Issues about View–Morphing

So far, it has been shown that view–morphing is a simple technique that can be used in the generation of correct views. However, to ensure its best performance there are a few issues that need to be considered.

3.3.1 Uniqueness and Monotonicity

There is a range of perspective views that can be uniquely determined from two or more basis views. A new view belongs to this range when it satisfies the *monotonicity* assumption.

Monotonicity dictates that all visible scene points appear in the same order along conjugate epipolar lines in the input images. Formally one can define this theorem as follows: given any points P and Q in the same epipolar plane, the angles θ_0 and θ_1 determined that the camera centres C_0 and C_1, respectively, must be nonzero and of equal sign.

If it is determined that all scene points in the two input images meet the monotonicity constraint, then all the intermediate images along the line C_0C_1 must also meet the constraint. This group of images is known as a *monotonic range* of view space.

An interesting and useful consequence of the monotonicity constraint is that intermediate images do not depend on the specific shape of the surfaces in the scene but only on the positions of their visible endpoints. Many interpolation methods require a dense correspondence of the input images. However in uniform regions, the correspondence is very difficult to determine. With the monotonicity constraint the only information needed is the one corresponding to image edges that are easy to identify in the images. Uniform regions are then interpolated to adapt to the new shape.

3.3.2 Visibility and Occlusions

When all objects and their salient features appear in both images and the input images are monotonic, the generated views are correct. However, when some features appear only in one of the input images, visibility problems arise in the form of *holes* or *folds*.

In the case where a feature appears only in image I_0 and then becomes occluded, this results in a *fold*: the mapping of multiple pixels to the same point in the intermediate image, causing an ambiguity. This problem can be easily solved if depth information of the scene can be obtained (usually by disparity estimation). With this information one can apply Z-buffer techniques to decide the visible surface in the resulting image.

Holes represent the opposite situation where a feature is only visible in the second image. These are more difficult to eliminate from image information alone. The most common methods to solve them are: setting a background colour, interpolating missing information from neighbouring pixels or, in the worst cases, resorting to the use of additional images.

3.4 N-View–Morphing

Up to now we have been focusing on a morph between two images. However, the range of images that can be generated from this configuration is very limited. Only the views found along the line joining the camera centres.

One of the great advantages of view–morphing is that the simple three–step algorithm can be extended to work with more input views simply by considering the closest views available and treating them as a series of two–camera system. With this configuration, the range of images that can be generated increases considerably.

In the case where three cameras are considered, the *strong monotonicity* constraint has to be met. This constraint specifies that for any points P and Q in the scene, the line PQ does not intersect the triangle $C_0C_1C_2$ determined by the three camera centres.

If strong monotonicity applies, the triangular region delimited by the three optical centres of the cameras constitutes the view space where novel views are guaranteed to be unique and have a correct perspective. Any intermediate view inside the triangle formed by the three input images can be synthesized by a series of two interpolations, as shown in the left side of Figure 3.3.

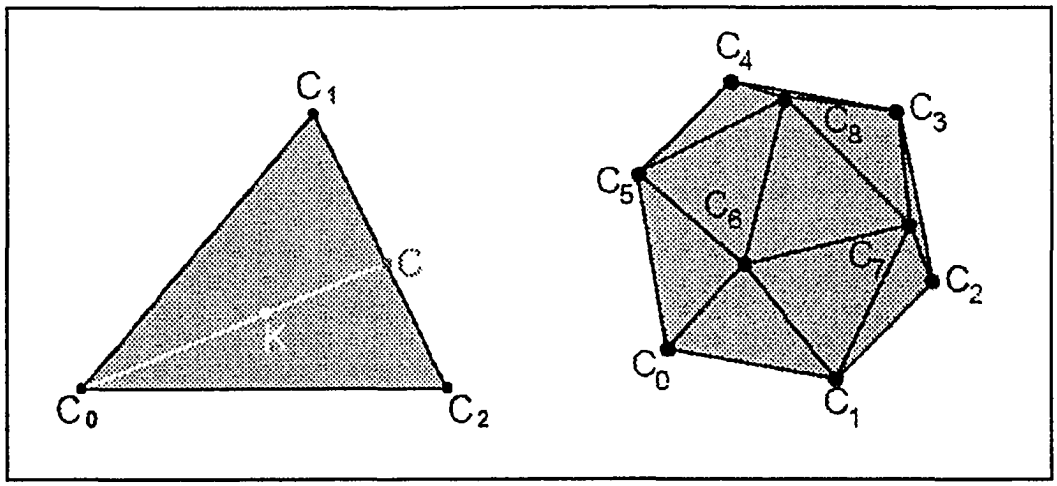


Figure 3.3: N-View–morphing: three-camera case (left) and multi-camera setting (right) [102].

The first view C is generated with an interpolation factor determined by the intersection point of the baseline of the two closest cameras (C_1 and C_2) and the perpendicular to the third camera (C_0). The second interpolation generates the desired view K from camera C_0 and virtual view C.

When more than three cameras exist, the desired view can be generated with three interpolations, as long as the desired viewpoint lies inside the convex hull of the camera centres [102].

Given a network of cameras, the closest three to the desired viewpoint are selected and the monotonicity constraint verified between every image pair (Figure 3.3 right).

3.5 Real-Time View–Morphing

In the context of this thesis, we are exploring the use of view–morphing for a tele–immersive system. For such an application, a completely automatic method is needed for the generation of new views. This method must perform in real-time in order to allow continuous interaction between the remote participants of the system.

Usually view–morphing requires the user to input common features between images. A modification of the standard view–morphing algorithm is proposed to avoid this user input in the selection of image features and their correspondences.

From the monotonicity constraint, it resulted that only endpoints of image intervals are needed for shape interpolation. Based on this result, we would like to determine what is the minimal set of image features needed for the morphing process in order to obtain realistic images that can be used for a tele–immersive application. Experimental analysis have shown that the contour of the person and few other features (e.g., eyes, nose, mouth locations) are enough to generate the intermediate views, providing adequate alignment of the features. These results will be analyzed in detail in Chapter 6.

Determining the minimal set of features needed is essential for the real–time performance of the view–morphing algorithm. Searching for a limited set of features in each image reduces considerably the amount of processing time. Because of the limited scope of a Master thesis, no attempt was made to implement this algorithm in real-time or to solve the problems associated with real-time implementation of feature tracking.

3.6 Discussion

One of the goals of this thesis was to define the framework of a future tele–immersive system from the point–of–view of architecture and algorithm. It was determined that a view–morphing algorithm can be adapted easily to solve the new viewpoint generation problem between two cameras and later between N cameras. We have demonstrated that this relatively simple algorithm has significant advantages over other methods. Toward the end of the chapter, we concluded that two important parameters are necessary to achieve this task. The first one is to find a way to robustly determine the values of the warp matrix *H*. The second one was to determine experimentally what is the minimal set of features necessary to do the morph and how to determine their locations in real-time.

In the next chapter, we will explore how to determine the warp matrix *H* for each camera from an exact photogrammetric calibration procedure that takes into account lens distortions. The exact estimation of the warp matrix *H* is critical since in order to guarantee monotonicity, the linear approximation of view–morphing must be satisfied at all times.

Chapter 4

Camera Calibration

Finding 3–D information from a series of images is a common task in computer vision applications and it can easily be solved when the images analyzed were taken from calibrated cameras.

The calibration process consists of estimating the intrinsic and extrinsic parameters of a camera.

In this chapter, we will consider a basic camera model and the steps involved in the calibration process.

4.1 Perspective Camera

The perspective camera is usually the model used for 3–D vision applications and it is derived from the ideal pinhole camera. Basically, this camera is defined by setting a centre of projection and a plane in which the image would be created: the retinal plane. In most cases this model and the projective transformation associated with it is enough. However, there are cases where some other effects must be taken into account in order to get accurate measurements such as lens aberrations (e.g., radial distortion).

To begin the description of how images are formed, it is necessary to describe the way projection takes place. The following equation shows the change of coordinates from the 3–D point $(X,\ Y,\ Z)$ to the corresponding image point $(u,\ v)$:

$$u = \frac{X}{Z} \qquad v = \frac{Y}{Z} . \qquad (4.1)$$

The previous equations can be expressed in homogeneous coordinates as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} . \qquad (4.2)$$

A system corresponding to Equation 4.2 defines the retinal plane ***R***, with principal point ***c*** perpendicular to the centre of projection ***C***.

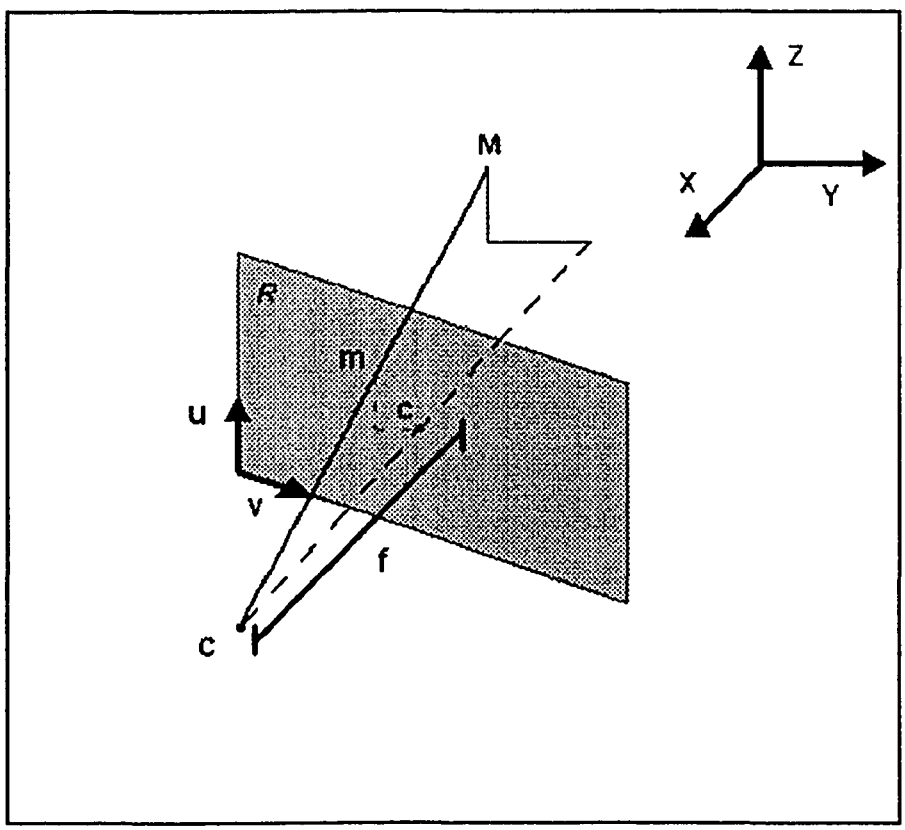


Figure 4.1: Perspective projection of world point M [89].

4.2 Calibration Parameters

In order to represent a real–world camera, there is a set of parameters to define [89] that can be grouped into two categories:

- Intrinsic parameters;
- Extrinsic parameters.

4.2.1 Intrinsic Parameters

This group of parameters is used to represent the internal characteristics of the camera: the optics and hardware elements that specify the corresponding projection that generates the images [37]. There are five intrinsic parameters:

1. The u-coordinate of the centre of projection c_u;
2. The v-coordinate of the centre of projection c_v;
3. The focal length f, expressed in pixels;
4. The aspect ratio a, between the pixels;
5. The angle between the optical axes s (skew factor).

The first two parameters define the centre of projection, that for most images, corresponds to the centre of the image. However, given that some inaccuracies may occur during the recording or digitizing process, some misalignment might be found through the calibration process.

For most cameras, the focal length is different from 1, as it was considered in the projective transformation defined above. Therefore this expression should be modified to consider the scaling effect caused by this change. To obtain the focal length value expressed in pixels, it is only necessary to divide the focal

length of the camera by the width of the image in the imaging surface (e.g., film or CCD array) and multiplying that by the width of the final image in pixels.

Finally, for the aspect ratio and angle between optical axes, there is the problem that coordinates in the image do not correspond with the physical coordinates in the retinal plane. The relation between both depends on the process through which images are digitized. In most cases it is enough to consider the aspect ratio to be 1.0 (when radially symmetric lens elements are used) and for good quality cameras, the angle between the optical axes and the CCD detector plane is equal to 90°.

All the modifications to the image coordinates resulting from the intrinsic parameters can be represented by the following matrix form:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_R \\ y_R \\ 1 \end{bmatrix} \qquad (4.3)$$

where $f_u = \frac{f}{p_u}$, $f_v = \frac{f}{p_v}$, p_u and p_v are the width and height of the pixels, c_u and c_v are the coordinates of the principal point, and s represents the skew factor due to non-rectangular pixels: $s = (\tan\alpha) f_v$; α is the skew angle between the optical axes.

This upper triangular matrix (4.3) is known as the **calibration matrix** K and describes the transformation from retinal coordinates to image coordinates:

$$m = Km_R . \qquad (4.4)$$

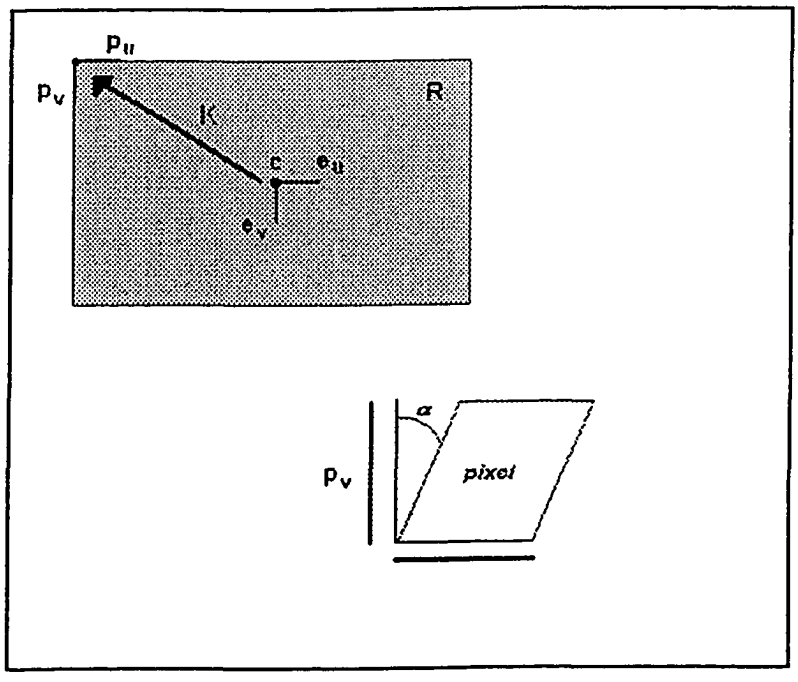


Figure 3.2: Transformation from retinal to image coordinates [89].

4.2.2 Extrinsic Parameters

To represent a real–world camera, its world position and orientation have to be taken into consideration too, especially when working with various images where camera motion information is needed to measure 3–D structure.

Camera motion can be represented in matrix form as follows:

$$M' = \begin{bmatrix} R^T & -R^T t \\ 0_3^T & 1 \end{bmatrix} M \qquad (4.5)$$

where R represents a rotation matrix and $t = [t_x \quad t_y \quad t_z]^T$ a translation vector.

4.2.3 Camera Projection Matrix

With the intrinsic and extrinsic parameters, the camera can be completely described and it is possible to write the projective transformation from a 3–D world to an image point. The final transformation combines the calibration parameters as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \approx \begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R^T & -R^T t \\ 0_3^T & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad (4.6)$$

which can be simplified to:

$$m \approx K\begin{bmatrix} R^T & -R^T t \end{bmatrix} M \qquad (4.7)$$

or

$$m \approx PM \qquad (4.8)$$

where the $3*4$ matrix P is the **camera projection matrix**.

4.3 Changes to the Model

Even though the pinhole camera model is quite good at representing the process to form an image, there are other things that modify the results from the equations above and have to be taken into consideration. In order to compensate those effects, some extra modifications to the model are done to get accurate measurements [37].

The first thing that can be noticed in real cameras is that light is not only received from a single point as in the pinhole case, but it is collected from points all across the surface of the lens of the camera.

One consequence of these non-paraxial rays, is that each point in an image is not the result of a single ray of light but a combination of all the rays converging from the lens front in a single point on the focal plane. Selecting an adequate aperture can attenuate this problem.

Another result from this property is that, given the fact that rays do not intersect in a single point, there may not be a mathematically precise principal point making it impossible to specify with absolute certainty the position in space from which an image was taken. However, this effect is most noticeable when using wide-angle lenses, thus it is possible to ignore this deviation in most cases.

Some other modifications to the basic camera model that affect the image formation are caused by failures in the optical system of real cameras [89]. Commonly known as aberrations (e.g., astigmatism, chromatic aberrations, spherical aberrations, etc.) these modifications are negligible under normal circumstances, but have to be taken into account for high-resolution work.

The most noticeable changes in the resulting images are caused by lens distortion, the cause of the transformation of real world straight lines into slightly curved lines in an image. This effect is usually known as radial distortion, given the fact that lenses are radially symmetric. Images suffering from radial distortion show a tendency to displace image points radially to or from the image centre due to the fact that objects at different angular distance from the lens axis undergo different magnifications.

To correct the effects of radial distortion, the centre of distortion (in most cases the principal point) must be recovered first. Afterwards, it is used in a radial transformation function that remaps pixels to straight lines. The equations below show how to obtain the undistorted coordinates $(u, \quad v)$ from the observed image coordinates $(u_0, \quad v_0)$:

$$u = u_0 + (u_0 - c_x)(K_1 r^2 + K_2 r^4 + ...) \qquad (4.9)$$

$$v = v_0 + (v_0 - c_y)(K_1 r^2 + K_2 r^4 + ...) \qquad (4.10)$$

where K_1 and K_2 are the first and second parameters of the radial distortion and r is defined as $r = (u_0 - c_u)^2 + (v_0 - c_v)^2$.

It is important to note that any modification to the focal length produces a change in the values of K_1 and K_2.

There exists the possibility that a camera may not have well aligned optic components, in which case, distortion patterns would be more complicated to model and correct.

4.4 Camera Calibration Method

Once the camera model has been established, it is possible to define a method to obtain the camera parameters from a set of corresponding 3–D world points and 2–D image points. Most of the times it is assumed that this 3–D to 2–D relation is linear, however when lens distortion exists, some non-linearity arise.

4.4.1 Basic Equations

Given a set of correspondences of 3–D points M_i and image points m_i, the idea is to find the 3*4 camera matrix P that satisfies $m_i = PM_i$ for all i. Assuming the matrix P is defined as shown below, it is easy to derive the equations to solve a linear system to find the eleven camera parameters excluding the four aberration parameters.

$$P = \begin{bmatrix} q_1 & q_{14} \\ q_2 & q_{24} \\ q_3 & q_{34} \end{bmatrix} \qquad (4.11)$$

where q_i, $i = 1,2,3$, is the vector of the first 3 elements of each row of P. For each point correspondence two equations are obtained:

$$q_1^T M_i - u_i q_3^T M_i + q_{14} - u_i q_{34} = 0 \qquad (4.12)$$

$$q_2^T M_i - v_i q_3^T M_i + q_{24} - v_i q_{34} = 0. \qquad (4.13)$$

Stacking the pairs of equations for each of the i points, the linear system defined is: $Aq = 0$, where A is the $2i*12$ matrix representing the 3–D and 2–D information and q is the $12*1$ vector of the camera parameters, defined up to a scale factor.

4.4.2 Linear and Non–linear Methods

From the context defined previously, one can notice that the matrix P has eleven degrees of freedom, thus only six points are needed to find an exact solution.

In the first case, when exactly six point correspondences are given and the rank of A is eleven, with no more than three coplanar world points, a linear method such as least squares can be used to solve the system. Unfortunately, once we include the four aberration parameters, the problem is no longer linear.

However, if the data is not exact and more than six point correspondences are provided, a non-linear mean square minimization algorithm can be used. In this situation some constraints have to be set for the minimization of $\|Aq\|$. The most commonly used are:

a) $\|q\| = 1$ where q is the vector containing all the entries of the matrix P;

b) $\|q_3\| = 1$ where q_3 is the vector formed by the first three entries in the last row of P.

Starting from the initial linear solution, a non-linear bundle adjustment program described in [39] is used to compute the aberration coefficients. The bundle adjustment program can deal with hill condition matrix created by the fact that most of the time the A matrix is sparse.

4.5 Implementing the Calibration Algorithm

To perform a calibration of any camera, the first thing we need to do is to define a set of 3–D world reference points using a pattern of well–defined features. Usually the model used for calibration is formed by two planes placed in a fixed angle where the surfaces are covered by a series of squares or circles with known coordinate positions. A picture of the calibration target used for this project is shown in Figure 4.3. It is composed of twelve circular targets of 2.54 cm in diameter spaced according to the target pattern illustrated in Figure 4.4.

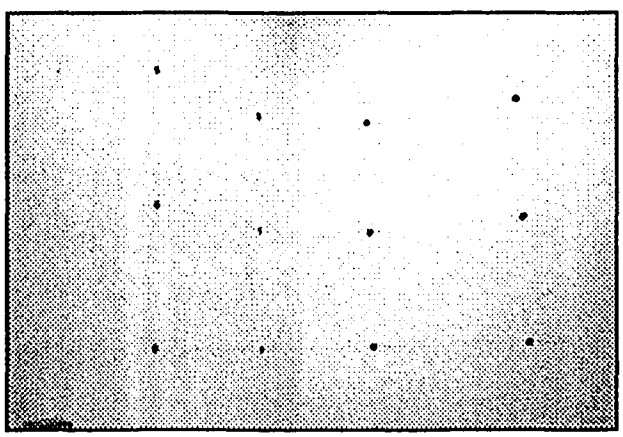


Figure 4.3: "Shape Capture" calibration apparatus [53].

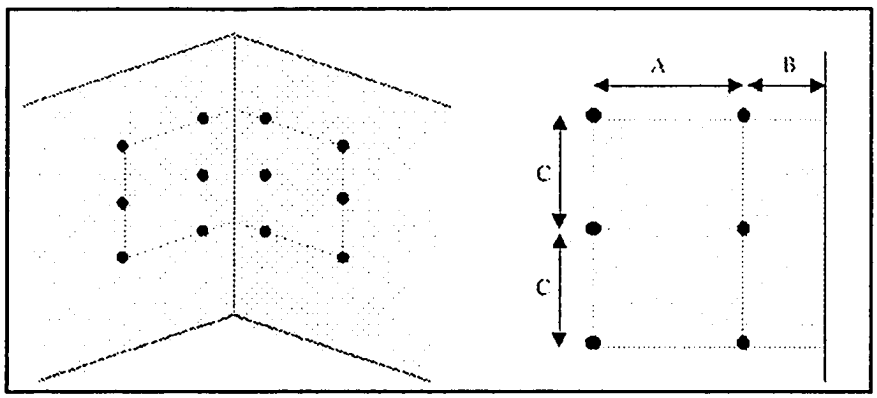


Figure 4.4: Design of Calibration Points [53].

Since we are taking the images from one to two meters, the following target design parameters was used for A at 0.50 m, B at 0.25 m, C at 0.30 m.

The calibration process can be summarized as follows:

- Acquire the images of the patterns with the cameras for which the calibration parameters need to be determined;
- Manually extract the targets from the images to determine their pixel coordinates;
- Compute the camera parameters using non-linear numerical methods such as a non-linear bundle adjustment program.

The camera calibration method used in this thesis can be found in [39]. Here the authors present a complete two–step method for calibrating a camera. The first step is to find the epipolar transformation by either Sturm's method or the use of the fundamental matrix. Then the intrinsic parameters of the camera are found from the equation of the absolute conic and the Kruppa [41, 76] equations. In order to do so, the algorithm starts by detecting the centroids of the targets illustrated in Figure 4.3 using a segmentation technique based on adaptive thresholding. Following this foreground/background segmentation, the position of

the target is determined at sub-pixel accuracy using a moment method [39], which is known to be very accurate and robust. From the computer target position in image space and real position of the target, the targets are identified manually and the solution to the non-linear calibration matrix problem including aberration is computed using a very stable non-linear bundled adjustment algorithm. To make sure this procedure is very accurate, we used a commercial photogrammetric package called *Shape Capture* [5]. The calibration procedure of this package is known to produce very accurate results that are in the order of one part in a million for five a Mega–pixels camera. It implements the so-called standard photogrammetric calibration model, an industry standard in the field of metrology. One can see in Figure 4.5 a typical output of the program for two cameras.

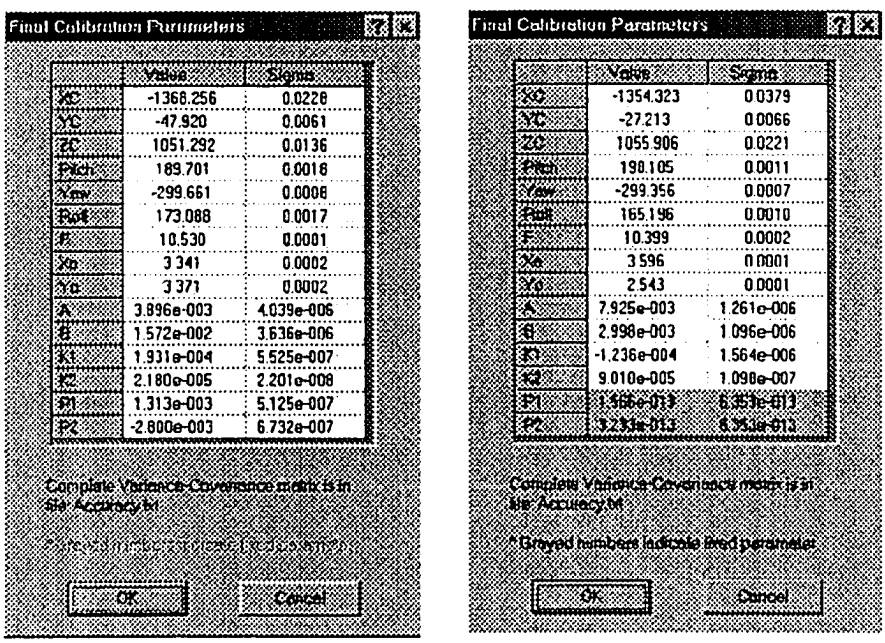


Figure 4.5: Typical output of Shape Capture calibration program.

4.6 Discussion

In this chapter, we have described a calibration procedure capable of computing the extrinsic and intrinsic parameters, including lens aberrations, necessary to automatically compute the warp matrix H that the view–morphing process need in order to compute to perform the morph in ideal linear conditions. We used Shape Capture, a professional commercial software to determine all these parameters.

One can find many papers in the literature on other methods to compute calibration parameters automatically from natural features, but usually at the expense of accuracy and algorithmic complexity. Using Shape Capture, the calibration process is simple and can be performed in less than five minutes as an initial setup of the system. Since the camera positions and settings do not change during a session, re-calibration need to be performed only when the cameras are physically moved.

Chapter 5

Feature Tracking System

Before view generation takes place, images from the network of cameras need to be processed to extract the essential information that the view–morphing algorithm requires. These operations include:

- Segmentation between foreground and background;
- Extraction and approximation of the foreground contours by linear segments;
- Identification, localization, and tracking of facial features such as the mouth, the nose, and the eyes.

This chapter describes the current state–of–the–art on how to perform those essential tasks from the point–of–view of computational efficiency and its ability to be automated. The exact real–time tracking of these features is key to the success of a real-time view–morphing subsystem.

In the context of this long–term work, these subsystems are currently being developed in the laboratory by other students. In this thesis, we only analyze the current state–of–the art of those subsystems in the context of tele–immersion.

5.1 Background Removal

In order to be able to generate new views and to integrate those views in the virtual meeting room, the tele–immersive system must first be localized and the various participants segmented from their background. In a situation where the scene viewed by the cameras remains constant in time and where the only

changes correspond to the participants being present, background removal techniques can be used to segment the participant from his background.

This process called background/foreground segmentation, or background removal, aims at segmenting changing regions in an image using robust image segmentation techniques. One of the simplest techniques to perform this segmentation process is to acquire an average background image in advance and to use it as a template to identify any important changes in the image that are greater than a predefined threshold. The main problem with this approach is that the background scene is not really constant, changes particularly in lighting conditions pose a significant problem for those techniques.

The main differences among various background removal techniques depends on how the background template is computed and how a pixel is classified to be part of the foreground or the background. In [77] a review of some common background subtraction models is presented along with a set of criteria to classify the performance of each technique. In addition, they propose a new segmentation algorithm that is more robust in the presence of illumination changes such as shadows and lighting. The segmentation process is defined as a 2–D multidimensional Gaussian classification problem in RGB space, using precalculated mean and standard deviation for each pixel. The background model is updated with every new frame but only for those pixels classified as background. This update process is responsible for a better robustness to ambient lighting, since slow changes in the environment lighting conditions can be integrated in the model.

Pixel classification is done by verifying that the colour value of each pixel lies within the ellipsoid of probability of the computed distribution. Whether a pixel is considered to be a foreground object or an illumination artefact depends on the value obtained from its brightness distortion value. This can be achieved by

computing a parameter that indicates how close the pixel colour is compared to its expected chromaticity value.

Another interesting method for background subtraction is the one presented in [111]. The novelty in this technique lies in the way the scene is modeled. The method analyses image properties in three levels: pixel level, region level, and frame level. Each level provides different information to the system and helps make the segmentation process more robust to miss classifications.

Other techniques try to improve the reliability of this segmentation process by adding more information to the classification process. For systems with more than two cameras, many authors [42] use range information to improve the robustness of lighting conditions using real-time Photogrammetry. Others [119] use range and intensity information such as the commercial system called Zcam. One can see in Figure 5.1(a) a picture of a desktop Zcam and in Figure 5.1(b) the broadcast version.

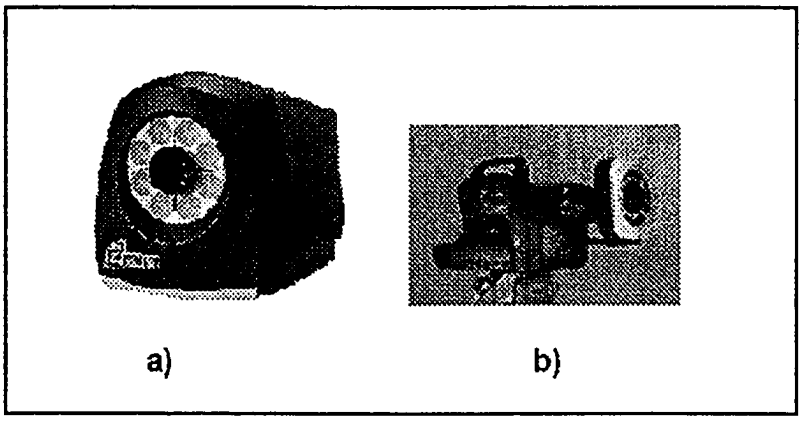


Figure 5.1: Zcam (a) Mini camera (b) Broadcast camera.

The operating principle of the Zcam is based on generating a "light wall" which has a proper width moving along the field–of–view (FOV). This "light wall" is generated by a square laser pulse of short duration (1 ns) having a field of illumination (FOI) equal to the FOV (Figure 5.2). Non-visible light is used in

order to not interfere with the video content. As the light wall hits the objects in the FOV, it is reflected back towards the camera carrying an imprint of the objects (Figure 5.2). This imprint contains all the information required for the reconstruction of the depth map. The depth information can then be extracted from the reflected deformed "wall," by adding a fast image shutter in front of the CCD chip, and blocking the incoming light as shown in Figure 5.2. The light collected by each pixel is then inversely proportional to the distance of the specific pixel. Since the reflectivity of any object varies, it is essential to compensate for this effect by a normalization procedure.

The normalized depth of a pixel $D(i, j)$ can be calculated by simply dividing the front portion pixel intensity $I_{front}(i, j)$ by the corresponding portion of the total intensity $I_{total}(i, j)$: $D(i, j) = I_{front}(i, j) / I_{total}(i, j)$. One can see in Figure 5.3 an example of a Zcam result. Using cameras such as the mini Zcam, real-time foreground/background can be achieved robustly and will be eventually used in the system. The AMMI laboratory is currently in the process of evaluating those cameras and if successful it will be incorporated in the prototype system.

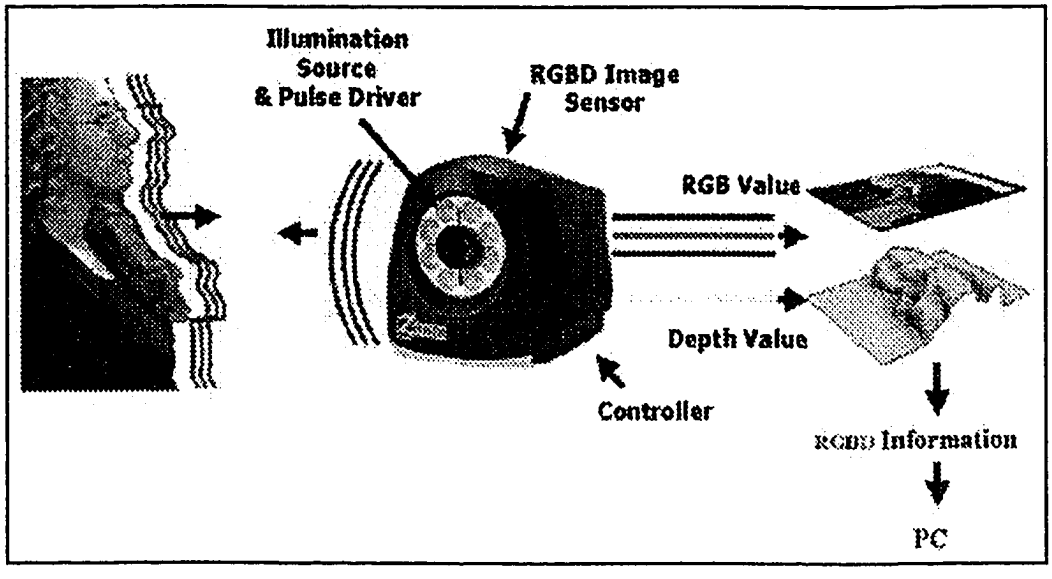


Figure 5.2: Principle of operation of the Zcam.

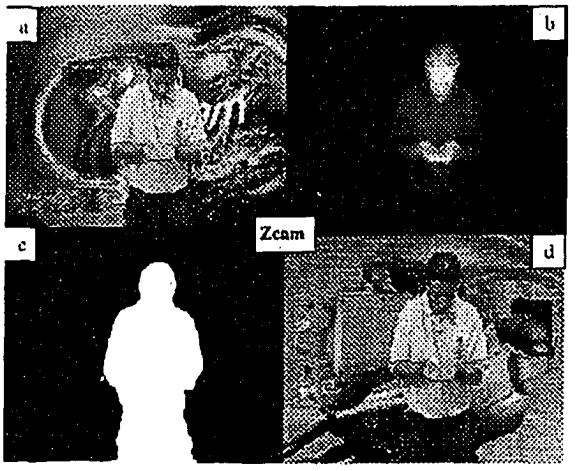


Figure 5.3: Zcam results: a) original image, b) depth map, c) foreground template, and (d) template integrated in new background.

5.2 Face Feature Tracking

Once the user has been segmented from the background by a background removal technique, a finer recognition of the user is needed in order to proceed with the view generation step of the system.

The second pre-processing step performed on the images is the localization and tracking of the user's face and in particular, facial features such as eyes, nose, and mouth, through the video sequence.

There have been many different approaches in computer vision trying to find a robust method for face–tracking. All the methods can be classified into two main categories:

- Global tracking;
- Local tracking.

5.2.1 Global Tracking

This type of method focuses on a general characteristic of the face such as skin colour, head geometry, and/or head motion. They analyze the whole image in order to find the face.

These methods have an advantage in that they are robust to head rotation and scale. However, their main drawback is their lack of precision. Focusing on big areas in the image does not provide pixel–level accuracy.

Among the most widely known techniques in this category of algorithm is based on the use of templates. The most general case represents an image as a two–dimensional array of intensity values and then compares this image with a template representing an average model of a face. Different modifications to this approach exist, depending on the pre-processing performed to the images, whether only one template is used or if there are different templates to represent a set of face poses or different features in the face, etc.

In the work by Brunelli and Poggio [25], a comparison between the performance of a global–tracking method using templates to the results obtained with a featured-based tracking system is presented. They found that template matching recognition performance is superior as well as simpler. Its main disadvantage being its memory requirements and speed, which limits its use for particular applications.

An interesting aspect of their work is the modifications they made to traditional template matching to make it more robust and improve its performance. A pre-processing step transforms the images into a map of the magnitude of the gradient. Then, when performing the matching they make use of a different template for each of the key features (eyes, nose, and mouth) instead of only one for the whole face. This aspect makes it an original approach in the sense

that they are somehow mixing elements of both local and global face tracking in order to improve the results.

Another method in this category is the one proposed by Zhu and Fujimura [124]. In their work, they propose the use of optical flow and depth constraints in order to estimate the 3–D position of the head. This method focuses on a teleconferencing type of application. They start by segmenting the user from the background, setting a range of depth values where the user is predicted to be. Once they have segmented the user, they localize the head separating it from the shoulders using a horizontal projection signature operation in conjunction with the depth information. Finally they obtain the head pose from the motion estimates previously calculated with optical flow.

A novel technique worth mentioning here is one using a Bayesian classifier to detect faces. Chengjun Liu [72] presents a method for multiple frontal face detection that is trained from an image database. Its novelty and robustness comes from the mixture of discriminating feature analysis, use of statistical models of both face and non-face images, and a Bayes classifier. Another important aspect of this approach is the accurate and complete information stored in the feature vectors it uses which combines the input image itself, its 1–D Harr wavelet representation, and its amplitude projections in both vertical and horizontal directions.

5.2.2 Local Tracking

This second group focuses on tracking a particular facial feature (eyes, eyebrows, mouth corners, nostrils). Methods in this group are great alternatives for applications requiring pixel accuracy.

One problem many of these approaches have is the need for expensive high–resolution cameras able to show and track the feature in detail. These kinds

of methods are not so robust to head motion, and face the problem of losing feature visibility during the tracking time period.

Besides the research done in tracking of facial features, each method explores the use of a different feature or group of features in order to obtain the best tracking possible.

One example is the work by Burl and Perona [27] that makes use of local feature detectors as well as a probabilistic model of the arrangement of features to identify planar objects such as faces. In this method, different feature detectors are used to track every feature according to its own characteristics. Although the authors claim to get good results with their probabilistic model and the kind of hypothesis they are generating, it is still a work in progress that does not take into consideration the errors of the feature detectors in their performance evaluation.

A big problem that a method like the one just mentioned previously faces is that trying to find and track many features in an image is not always the optimal solution. Sometimes adding more features to the image analysis introduces more uncertainty in the results obtained from the detection depending on how accurately each feature can be tracked.

To decrease the ambiguity in the tracking of faces, it is suggested to track only one feature whenever possible. In a later chapter of this thesis it will be shown that this is exactly what works for our tele–immersive system. There are key features which correct tracking can determine the success of the whole system and the realism of the immersion feeling.

Focusing on the videoconference/tele-immersion context, important features to track are the eyes (that provide a lot of information to interlocutors in a conversation) and either the nose or the mouth to increase the robustness of the

tracking, providing additional information for correct face alignment during the view–generation process.

Many different methods aim to track the eyes and estimate the gaze direction of the user of a system. The interest in this kind of tracking lies in the numerous applications eye–tracking could be used for to improve human computer interaction. We can mention as an example, the work by Talmi and Liu [108] that uses three cameras to track the eyes and estimates the gaze direction in order to present adequate perspective images to the user of an auto–stereoscopic display. Another interesting application of this kind of tracking is in monitoring user awareness as in the system presented in [59]. Here some of the eye parameters recovered from the processing are used to measure the awareness of a driver.

The methods to track and segment the eye from a video sequence differ principally in the different situations they are robust enough to handle. In [109] the head is extracted from the image by colour segmentation. Then the position of the eyes is estimated by a corner detector and from there two hypotheses of the actual iris position are obtained: one using region–growing of the pupil and applying template matching, the second by a Hough transformation. Both hypotheses are combined, and with the position of the eye corners the gaze direction is successfully estimated.

A very robust and easy to implement method is the one proposed by Zhu *et al.* [125]. In their work, they use an array of infrared LED that helps in the localization of the pupil in the image. With the infrared light of two different LED rings turned on and off consecutively, images of a bright and dark pupil are acquired. Both pupil images are subtracted, and connected components analysis is applied to the resulting blobs in order to find those representing the eyes based on their geometric shapes. One of the main benefits of this method, besides its robustness and accuracy, is that it is tolerant to different light conditions.

In the case of the nose, not much attention has been put into the robustness and simplicity to track this feature. Efforts in this direction have followed the standard and tracked the nose corners (nostrils) that are susceptible to occlusions.

Gorodnichy [46] presents a novel method that takes advantage of the characteristics of the nose: uniqueness, prominence, convex shape, and visibility at all times during user interaction with a computer system. The author presents encouraging results to follow this line of research in order to improve face tracking performance.

In his work, instead of representing the nose as a static point to track, the feature he tracks is the point closest to the camera. This definition of the nose feature allows for a wider tracking range given that the nose tip appears, even when the rotation of the head with respect to the camera is very large. The method starts by learning a template of the nose. Later, the template is used to track the nose in the following frames taking into consideration the previous position to determine the search area.

Throughout this section, some alternatives for face–tracking, and particularly eye– and nose–tracking were discussed. Although a lot of work in this area has been done, facial–tracking is still an unsolved problem with room for improvement in many aspects. A possible solution for face–tracking could be a combination of characteristics of both groups (global and local tracking) as it has already been tried by some research groups. However, the problem lies in the increasing complexity of the computational power needed that can make real–time tracking difficult. If in the future prototype, the decision is made to use the Zcam as the main camera system, more advanced algorithms based on the analysis of range and colour signals may yield results that may solve these problems.

5.3 Discussion

In this chapter, we reviewed the current state–of–the–art on how to robustly extract a participant to a meeting from its background. We have shown that by using a Zcam it is possible to segment the contour of a participant robustly and in real–time. We have also shown that a participant can be easily extracted from his/her background using as simple classification scheme based on colour and range information.

We then analyzed the literature on facial feature tracking and detection and concluded that it is a hard problem that requires new ideas. One possible new idea would be to use the range information of the Zcam as a way to improve the view–morphing problem by finding correspondence between key features in the range signals. This may open the door to new algorithms that may allow real–time tracking and view–morphing. In the next chapter, we will analyze in great detail, how the view–morphing process works and try to determine what are the minimal key facial features necessary to get a good morph.

Chapter 6

View–Morphing for Tele-Immersion: Implementation and Results

Now that the theory behind the main components of our tele–immersive system has been explained, let us analyze how to integrate those elements into a prototype tele–immersive system.

As mentioned previously, the implementation of the whole tele–immersive system is beyond the scope of this thesis. Where experimental results have been obtained, an in–depth discussion will be presented. Where results are not available, a discussion on how those elements should be integrated will be presented.

6.1 Hardware Implementation

The prototype desktop tele–immersive system we built is composed of two synchronized video cameras mounted on an 18-inch DTI [2] auto–stereo display screen. This configuration has the advantage of allowing the user to perceive stereo without wearing glasses. Currently the cameras are connected to an analog side by side multiplexer that combines video signals from the two cameras in a single frame of 640 x 480 pixels.

Audio is also an important part of the system, so a microphone and a pair of speakers are also part of this setting. The audio signal is associated with each video frame, ensuring synchronicity. An H.323 hardware codec is used to code both audio and video signals and to send them through the network to the remote participant at a rate of 1.5 Mb/s.

In the laboratory two of those systems were constructed to allow testing. Both systems have exactly the same hardware configuration. In Figure 6.1, one can see a picture of one of the desktop systems.

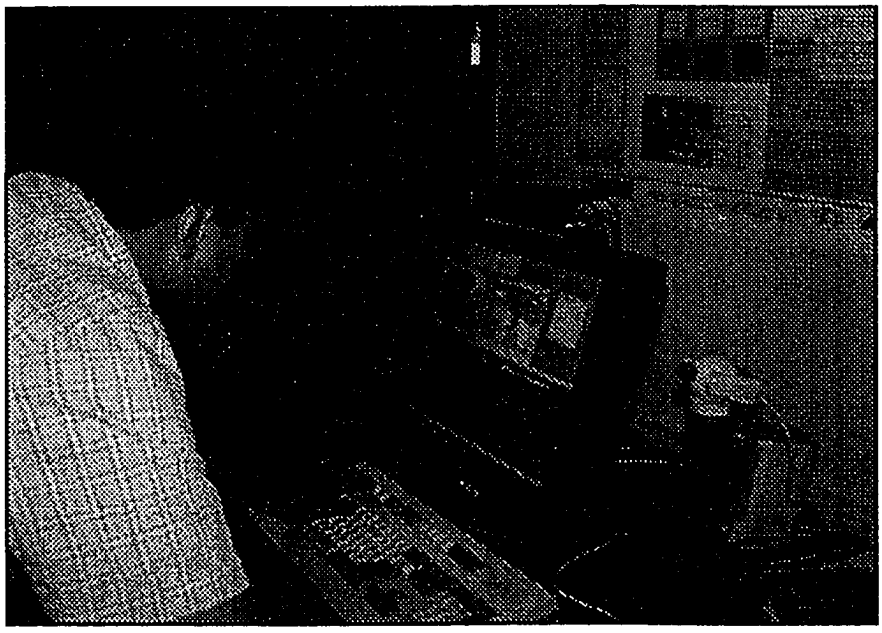


Figure 6.1: Desktop tele–immersive system.

6.2 Software Implementation

In terms of software, the data processing performed by each participant's computer is divided into three main processes. These processes are:

- Camera Calibration;
- Transmission;
- Reception.

In each system the transmission and reception processes are performed in parallel. If the system has only one processor, the processes simply are different threads running on one CPU.

The calibration of the cameras is only performed if there are changes in the physical parameters of the camera such as inter–camera distance, focal length, or zoom settings.

In this section, an overview of the calibration, transmission, and reception processes is presented. A prototype of the transmission and reception process was implemented last summer by other members of the laboratory and is currently working. In this thesis we only present how this process works and how it can be integrated in the overall architecture of the system.

6.2.1 Camera Calibration

For our current experiments, the calibration was performed using Shape Capture 4.0 [5]. As mentioned in Chapter 3, this software uses a bundle adjustment algorithm to obtain all fifteen extrinsic and intrinsic camera parameters.

The calibration process consists of showing a network of targets to the cameras as discussed in Chapter 3. The images obtained from the two cameras are then saved and loaded in the Shape Capture software. Following manual segmentation and identification of each target, the calibration parameters are computed automatically. The entire procedure takes about five minutes from beginning to end. The resulting parameters are then saved in an ASCII file. These parameters are a unique descriptor of the camera pair at each site.

6.2.2 Transmission Process

The transmission algorithm implemented in one thread is the following:

- Initialize H.323 transmission to communicate with the other participants. Once the H.323 process is running, the user is able to either initiate communication with others using a similar system or receive invitations from them in order to start a virtual meeting.
- Initialize system by camera calibration.
    - Run Shape Capture software to obtain the calibration parameters.
    - Using the calibration parameters, including lens distortion, compute the warping matrix H for each camera.
- Transmit warping matrices to the remote participants.
- Use a modified version of the method in [118] to create a background model of the scene.
    - Accumulate video frames of the background image during ten seconds. This time allows the model to include some illumination changes in the environment.
    - Create a background template for each camera from the statistics extracted.
- Rectify background templates with the corresponding warping matrices.

- For each frame after the user enters the scene:
    - Pre-warp each image of the user with the corresponding warping matrix to obtain a parallel stereo pair of the user.
    - Segment foreground/background in each image using the background templates. Set the background pixels to black, leaving only colour information of the user. Update the background template to reflect changes in the scene.

- Detect the largest connected component, representing the user, and obtain its contour. Simplified user's contour using the smallest amount of line segments possible for its representation.
- Match feature points on the contour and those corresponding to facial features in both images.
- Compute the disparity for each of the feature points by a simple correlation method along epipolar lines.
- Detect eyes and nose positions using facial feature trackers like the ones described in Chapter 4.
- Label pixels as background, foreground and feature points to create a label map.
- Compute the location of the participant relative to the common VR world using the disparity values along the contour. Position changes of the user in the real world must be reflected in the virtual world.
- Broadcast user position, warped images and label map to remote participants using H.323

- Loop until system session ends. The session ends when either the user decides to close his/her application or all remote participants leave the session.

6.2.3 Reception Process

The reception algorithm implemented for the second thread is:

- Initialize H.323 connection with remote participants.
- Read 3–D model of the shared VR meeting room. The participants meet in a virtual world where they are represented by avatars. After a session is established between users, the virtual world in which they interact has to be determined.

- Read warping matrices and initial position of each participant in the virtual world. In this step the necessary information to populate and update the virtual world is obtained.

- Main rendering loop (population and update of virtual world)
  - For each participant:
    - Read and decode rectified participant image pair and the corresponding label map.
    - Read participant's position and update his/her quad inside the virtual world. Reflect participant's change of position in the real world in the virtual world.
    - Establish correspondences between features of the stereo pairs using the label map.
    - Linearly interpolate corresponding edges one scanline at a time to obtain warped images of the remote participant. Warped images are generated for the inter–ocular distance of local user.
    - Postwarp generated stereo pair and texture map it onto participant's quad.
  - Perform scene rendering. Render virtual world with the texture–mapped stereo avatars representing each of the participants in the meeting.
- Loop until session ends.

6.2.4 Image Warping

For the first and third steps of the view–morphing algorithm, each image has to be warped to modify its orientation.

Image rectification can be done in different ways, the most widely used technique being the one proposed by Seitz [103]. This technique determines the uncalibrated value of the fundamental matrix by asking users to specify at least five

control points. The method is not really easy to automate and previous experiments have shown that pointing errors in the order of one to two pixels may result in unstable determination of this matrix.

For our work, we use a simple technique to perform image warping. First images are texture–mapped onto polygons. Then, a warping matrix is applied to each of these polygons to change their orientation as needed. In the current implementation the warping matrices are determined from the calibration parameters of each camera.

6.2.5 Background Segmentation

This step is a crucial step to simplify the processing tasks in the view generation. We work with a modified version of the algorithm in [118].

First the background model is built by capturing images of the scene without the user for one second and generating the statistics about the pixel colours. From the experiments we performed, the information from thirty frames is generally enough to obtain reliable statistics that include small variations in illumination conditions.

Once the background template is ready, it is used to detect when the user enters the scene: a big change is detected in the new frame. At this point, the segmentation starts by building a connected component representing the user. The background model is also updated to include changes to the scene through time.

User segmentation is done by a Bayesian classifier that uses the statistics of the background colours to assign a pixel with either a background or foreground label. A diagram of this process is shown in Figure 6.2 A more detailed description of the algorithm can be found in [24].

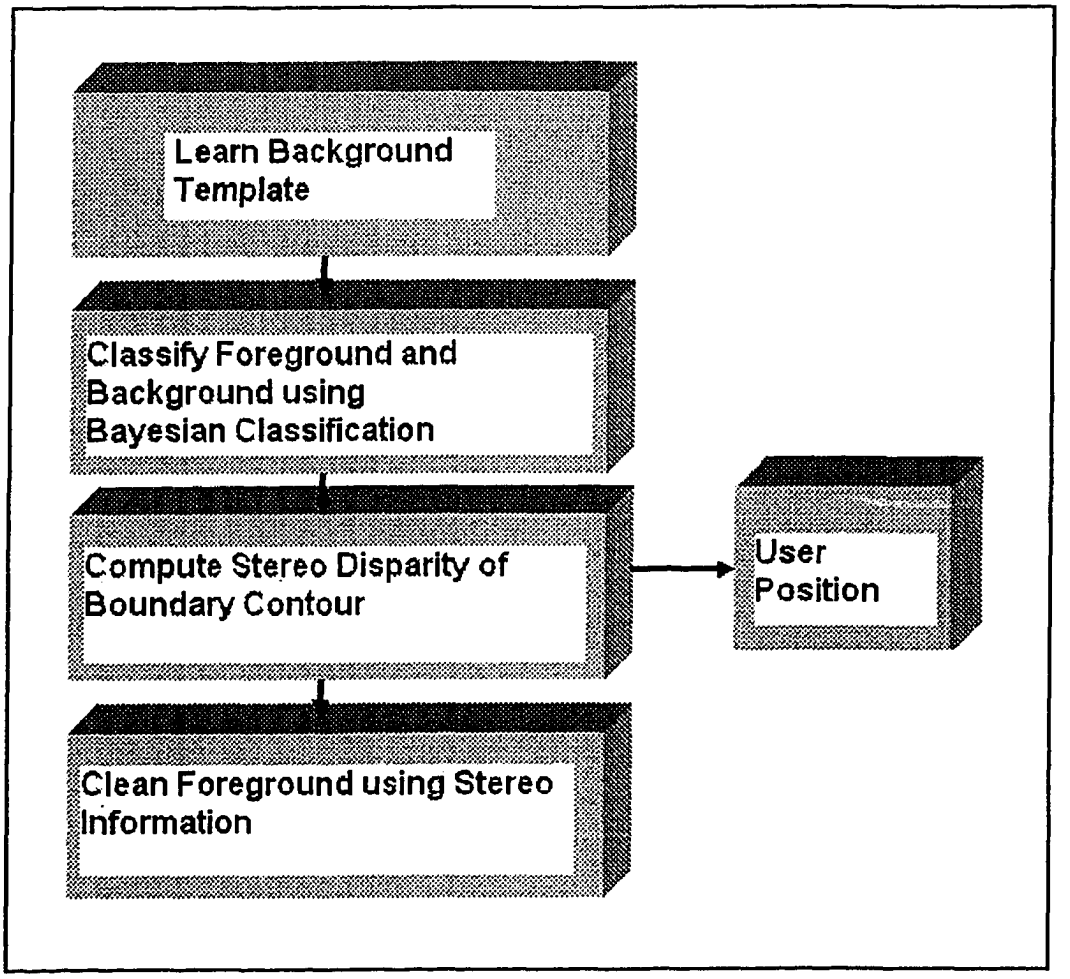


Figure 6.2: Foreground/background segmentation process.

6.2.6 Contour Simplification

So far our experiments have been done with contour segments specified by hand to have control over the precise position of the segments being used.

For future experiments the idea is to implement a piecewise linear approximation algorithm. This algorithm, given the connected component representing the user, simplifies the complexity of its contour using the smallest amount of straight–line segments to represent it.

6.2.7 Facial Features Detection

Facial features have been also precisely chosen by hand for our experiments.

In the future, we are planning to implement feature trackers like those proposed in [46, 47] for nose and eye tracking in the images. For these trackers to work properly feature templates are created by hand in advance. The template is then used to find the facial feature inside the connected component.

Another approach we want to study for this task is the use of infrared information. A method like that in [125] could increase the robustness of the feature tracking significantly with respect to those trackers based only on colour information alone.

6.2.8 Edge Correspondence

For this task, we apply a simple stereo correlation technique along epipolar lines. A review of different methods can be found in [100].

The feature points to match are the endpoints of each of the line segments defining the contour of the user, as well as those defining his facial features.

As there are just a few feature points, and the disparity values allowed for foreground pixels are restricted by the camera configuration, the correspondence of all points is quickly determined.

We used the same Simple Area-based Approach proposed in [117]. The algorithm is as follows:

- For each pair of corresponding scanlines in the label maps:
    - For each feature point on the left scanline:
        - Use the disparity limit to determine a search area in the right scanline.
        - Match the left feature point with its corresponding feature point in the right scanline, within the search area. To determine correspondence use a dissimilarity measure (e.g. Sum of Squared Differences).

This process is repeated exchanging the left and right label maps. This step ensures that the correspondences obtained are unique, disposing of those found only in one of the executions.

6.2.9 View–morphing Algorithm

All our experiments have been done using the feature–based image metamorphosis technique [21] since the line segments representing the contour and facial features are used for the interpolation.

One of the main advantages that this approach provides is the tolerance to some error in the localization of features in the image. Given that this technique assigns weights to the distance from a pixel to a feature, even if the feature is not perfectly localized, the result can still be satisfactory.

The morph consists in the linear interpolation of the edges representing the user, the mapping of the regions inside the edges to their new position and a cross-dissolve of the pixel colours from the input images. The algorithm for corresponding scanlines in images I_{left} and I_{right} is described below.

For each pair of corresponding scanlines:

- For each corresponding edge pixel pair (e, e'):
  - Linearly interpolate the edges to their new position:

$$e_{warped} = e'_{warped} = (1-s)e + se' \qquad (6.1)$$

where $s \in [0,1]$ corresponds to the interpolation factor.

- Map the regions between the edges in I_{left} and I_{right} to the regions between the newly interpolated edges by an 1–D forward resampling function [115] generating warped images I_{Wleft} and I_{Wright}.

Each pixel contributes to a pixel in the new regions either completely or partially depending on whether the new area is larger or smaller than the original ones (Figure 6.3).

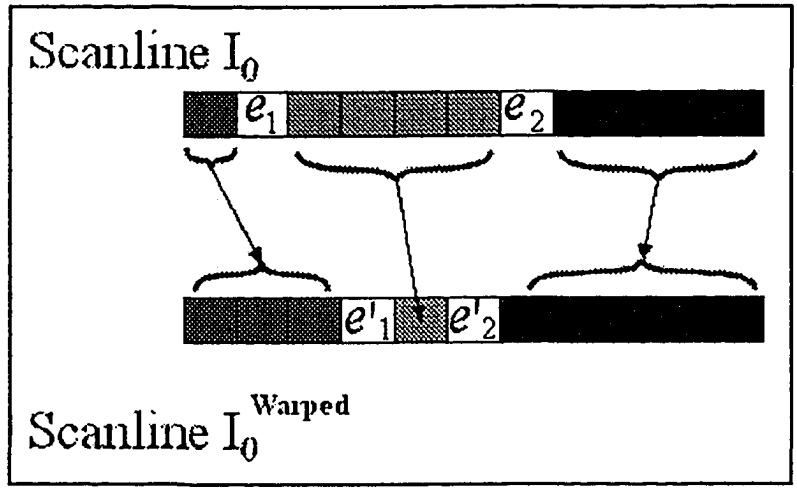


Figure 6.3: 1D warping of a scanline.

- Cross-dissolve the warped images by the interpolation of their intensities:

$$Intensity(I_s) = (1-s)Intensity(I_{Wleft}) + sIntensity(I_{Wright}). \qquad (6.2)$$

6.3 Experimental Results

For our experiments, we used both computer–generated images and real–life images, focusing on the latter and particularly on head shots for a videoconference setting. We used BMP image files with a resolution of 512 x 512 pixels.

The key aspects that we were trying to prove in this experiment were the realism of the images generated with the implementation of the view–morphing algorithm and the possibility to define a minimal feature set to be used in a tele–immersive application.

To verify the results obtained, image subtraction and colour histograms of the generated views were used to compare novel images with their ground truth counterpart whenever possible.

6.3.1 Image Subtraction

This first test is used to measure the number of pixels that have a different colour in the generated view with respect the one expected from the real image.

For this test, we have a pair of images captured at two times the average inter–ocular distance and an image captured at one inter–ocular distance. The morphed image resulting from the view morph at inter–ocular distance is then subtracted from its real counterpart. This process is shown below.

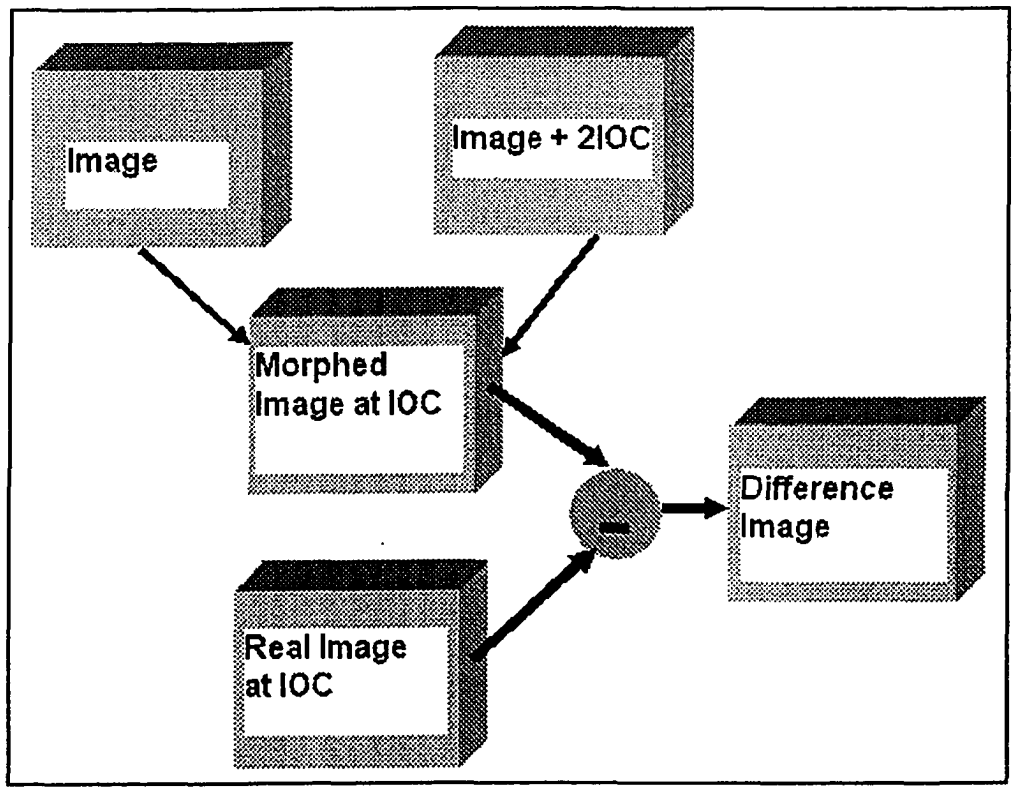
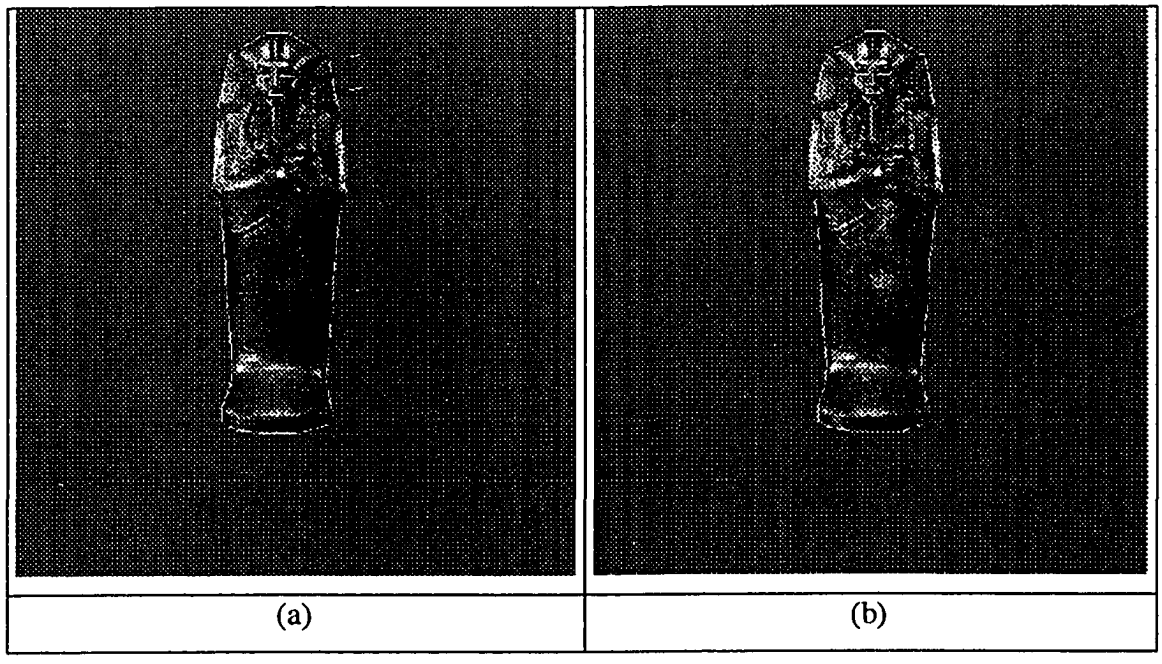


Figure 6.4: Image subtraction process.

A stereo pair of images of an Egyptian coffin used in this test is shown in Figure 6.5 In these images white line segments were superimposed to show the features used for the morphing process.



(a) (b)

Figure 6.5: Original (a) left and (b) right images of an Egyptian coffin.

It can be seen in the previous images that the contour was significantly simplified, and that only a few features in the face were considered for the morph. This selection is done in order to have fewer line segments to process, thus accelerating the generation of the morphed view. More details on the reasons why a particular feature was selected will be discussed in the following sections when we will analyze the of feature location on the view-morph results.

The resulting morphed image was then subtracted from the original image at inter–ocular distance. Figure 6.6(a) shows the view morphed image and Figure 6.6(b) its difference relative to the real image.

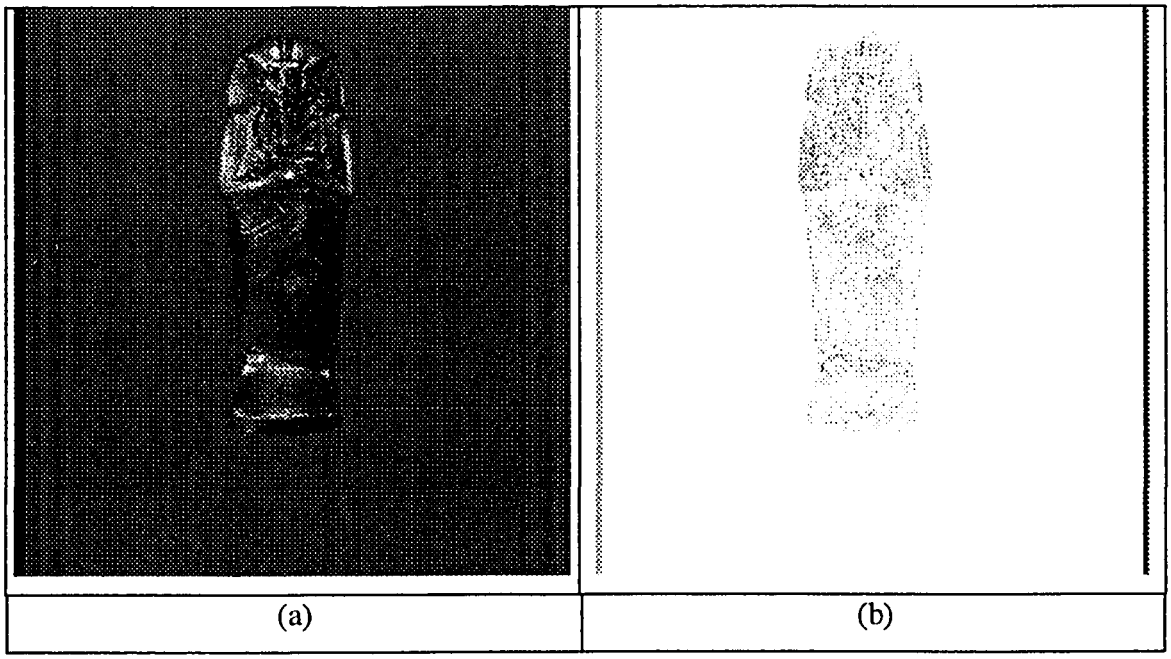


Figure 6.6: View–morphing results: (a) View morphed coffin (b) differences from the real image.

This image pair is particularly useful for the experiments due to its similarity to a human shape.

When observing the difference between images more closely, one could conclude that the view–morph results are not very good, given that 14% of the pixels in the image have different values from the ones expected. These differences consider only foreground pixels and represent any color variation with respect to the original image (threshold = 0). However, we have to remember that one of the restrictions of view–morphing is that it only works for lambertian surfaces and this image is of a specular statue.

On the other hand, looking at the morphed image, it is possible to appreciate that the result is very realistic. The generated view not only preserves the alignment of the contour segments used for the morphing process but also clearly shows the inner edges that were not matched in the morphing process.

of the contour segments used for the morphing process but also clearly shows the inner edges that were not matched in the morphing process.

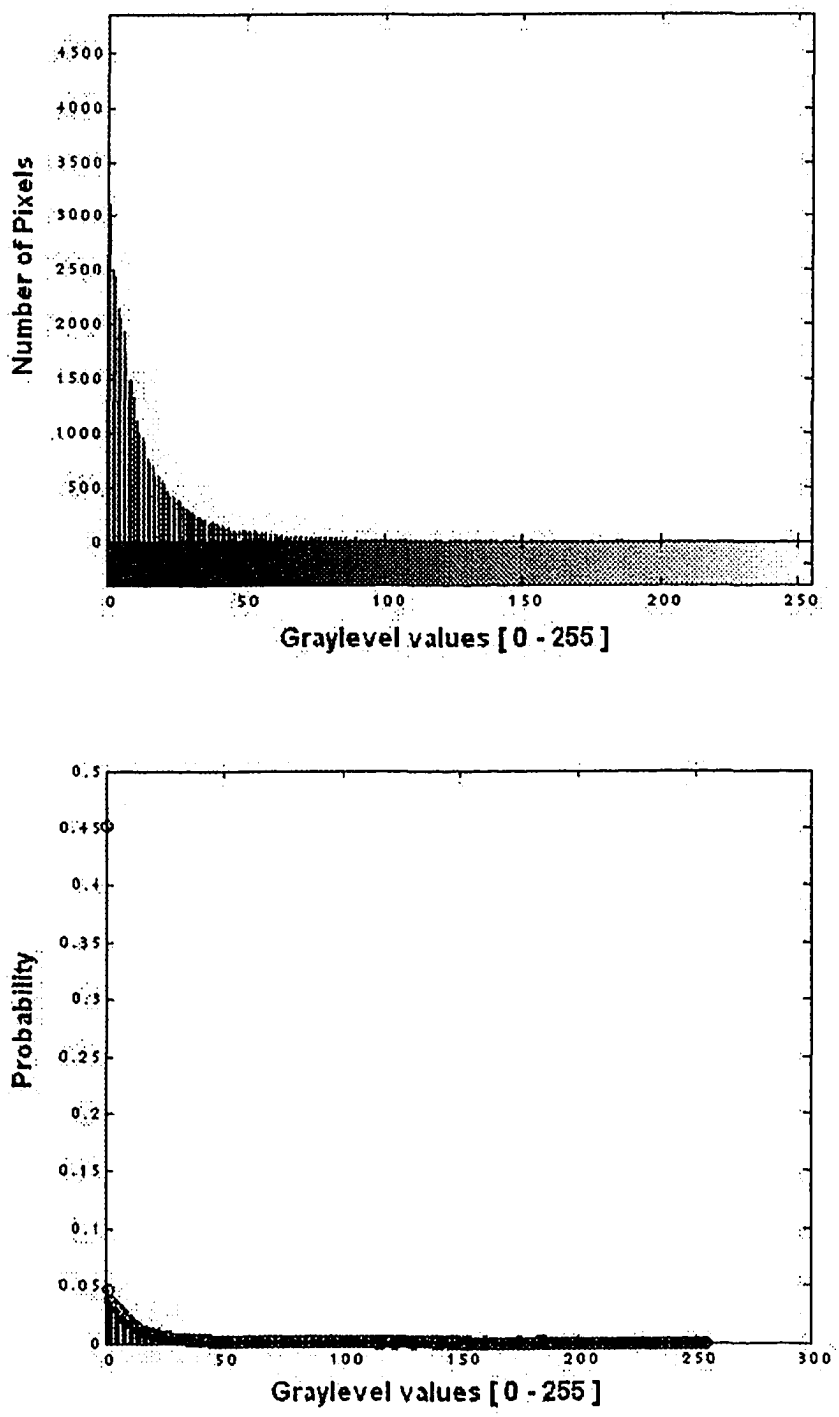


Figure 6.7: Coffin results: (a) Color changes (b) Probabilities.

For our system, we need to generate realistic representations of people interacting with each other. The next image pair, shows some of the results obtained using view–morphing for a tele–immersive application.

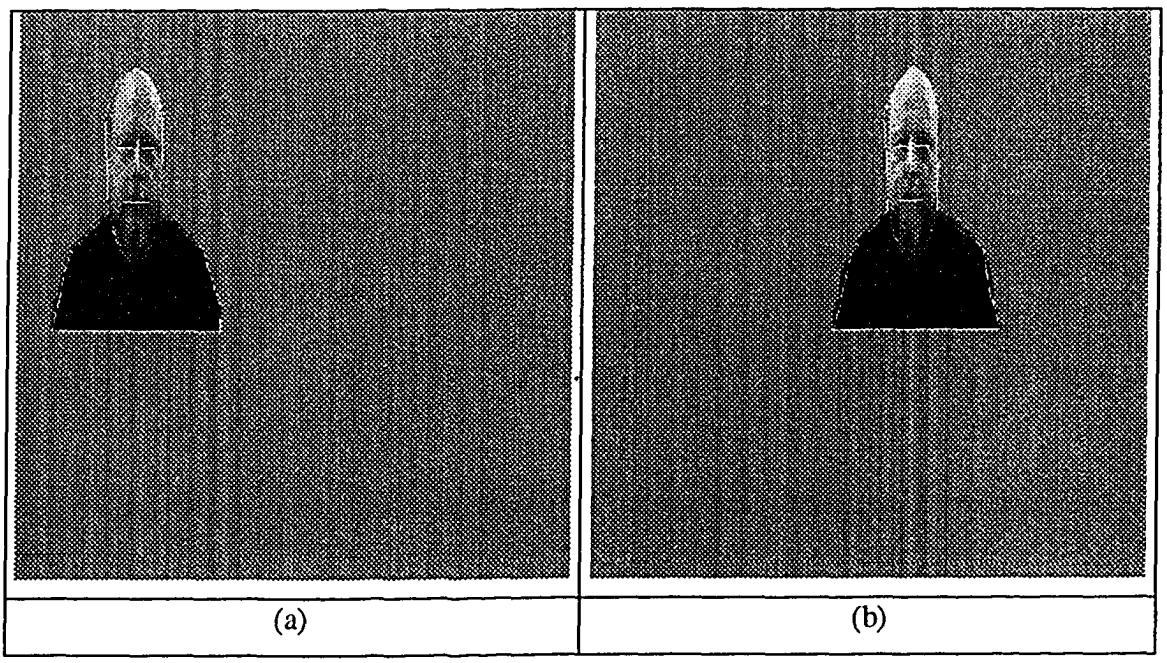


Figure 6.8: Stereo pair of Pierre: (a) Left and (b) Right.

The images in Figure 6.8 are a stereo pair captured by the cameras of the prototype tele–immersive system. The features used for the morphing processing are shown as white line segments.

For these images, we did not have a ground truth image corresponding to the middle view. What we tested in this case were the results obtained with different sets of features in order to find the optimum set. Figure 6.9 shows the morphed image obtained with the same feature set depicted in the images in Figure 6.8. Details on the influence on view–morphing of those features will presented in the next section.

of features in order to find the optimum set. Figure 6.9 shows the morphed image obtained with the same feature set depicted in the images in Figure 6.8. Details on the influence on view–morphing of those features will presented in the next section.

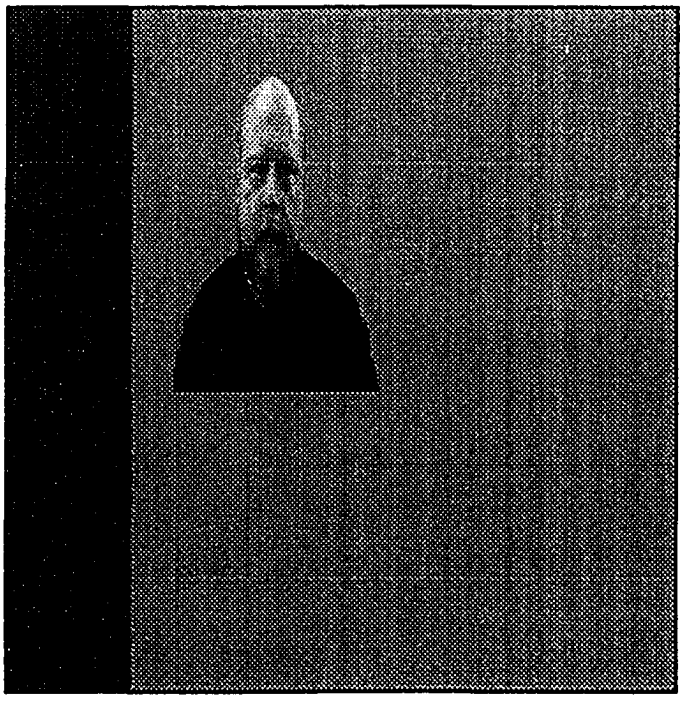


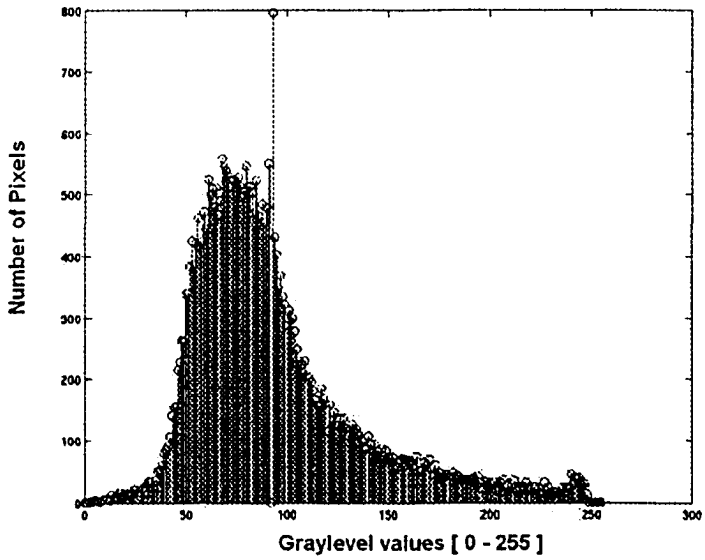
Figure 6.9: Morphed image of Pierre.

One can see in Figure 6.9 that a realistic image of a person can be generated using view–morphing with a simplified set of features. In this case using only a simple contour and the positions of the eyes, nose, and mouth was enough to obtain life–like results with correct alignment of all the important facial features.

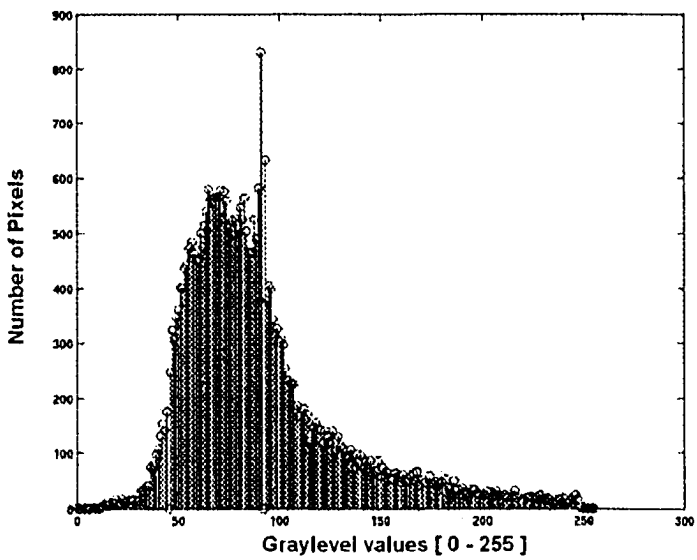
6.3.2 Comparison Using Colour Histograms

As a second part of the experiment, we analyzed the colour histograms of the generated views. The colour histogram shows the distribution of pixel values in the images. For this analysis, a graylevel version of the morphed images was used by adding RGB channels.

Results of this analysis show that the colour differences are very small and in most cases are directly related to illumination characteristics of the images. In Figure 6.10, we show the colour histograms of the original Egyptian coffin and its view–morphed counterpart and in 6.11 those of the stereo pair of Pierre.

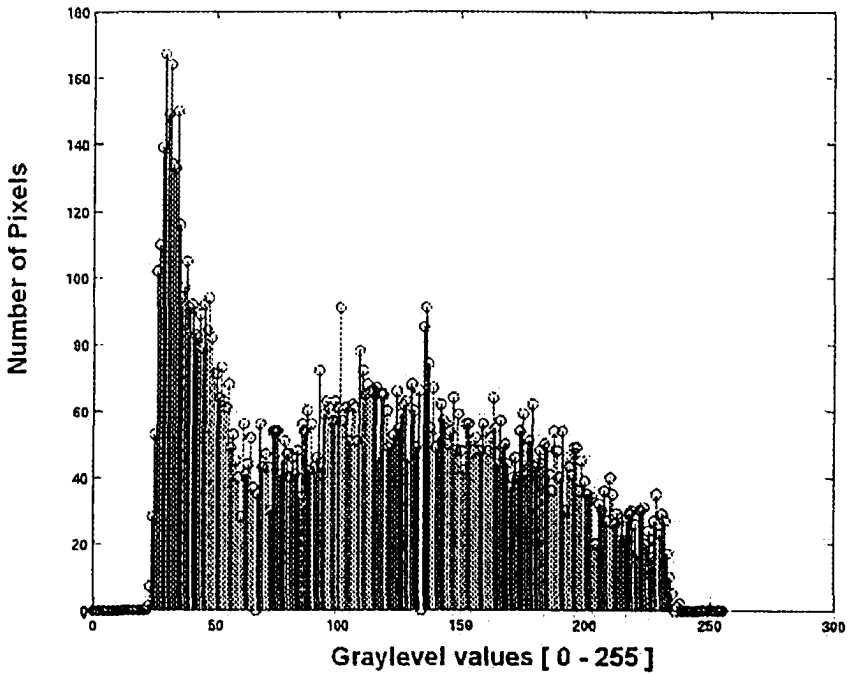


(a) Original image histogram

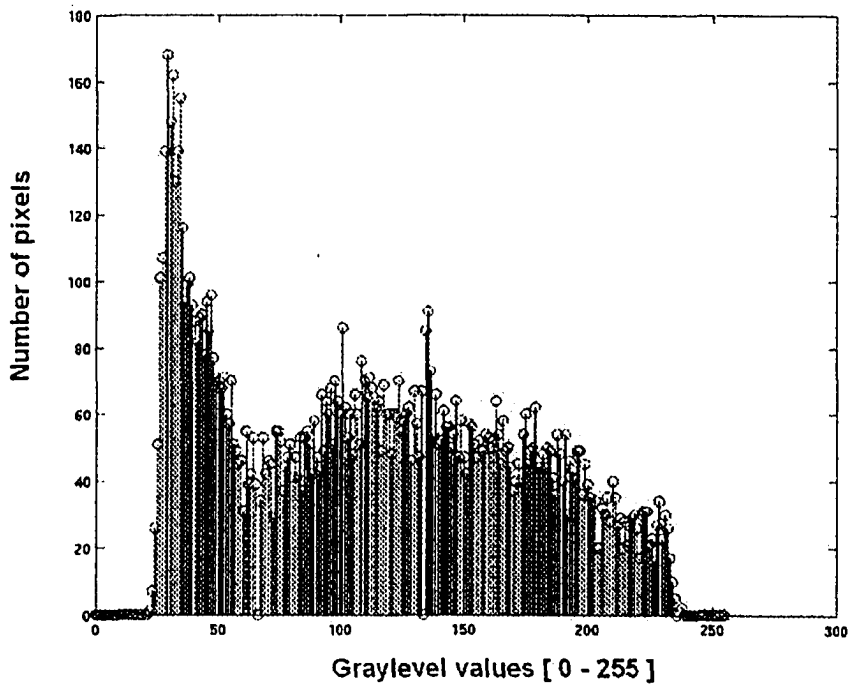


(b) Morphed image histogram

Figure 6.10: Colour histograms of (a) original and (b) morphed coffin images.



(a) Original image histogram



(b) Morphed image histogram

Figure 6.11: Colour histograms of (a) original and (b) morphed Pierre images.

Both histograms show that the colour distribution in the images is preserved fairly well during the morphing process. In order to compare more clearly the differences, the values of the background colour (known *a priori*) were eliminated from the graphs, as they represented around 50% of the pixels in the images.

One can notice in both histogram sets, the coffin and the stereo pair, that the general shape of the distribution is the same but that the morphed image is slightly smoother than the original. This can be explained by a blurring effect due to illumination effects and the cross-dissolving process.

6.3.3 Importance and Number of Features

Throughout this chapter some view–morphed results were presented. However, we have not talked yet in detail how to select particular features for the morphing process.

Since the core part of the view–morphing algorithm is the interpolation of the specified features, its performance depends directly on the amount of features selected as well as the morphing method used.

For our implementation, we used the technique in [21] for the morphing step. This method specifies each feature as a line segment in each of the input images, and interpolates their position to generate the intermediate image. The time to generate an intermediate view is thus completely dependent on the morphing step and the number of features used.

However, since realism is a key requirement for our system, in order to really feel immersed, the features used have to be carefully selected in order to obtain both fast processing and high quality morph.

For the tele–immersive system setting, we generated intermediate views with different sets of features, trying to find the one with the least number of features possible and the best results.

Methods based on edge detection have a tendency to find a large number of edges in pictures [117]. Having in mind our need to decrease the number of edges and to avoid the extra processing time that an edge detector adds to the system, we took advantage of the information already obtained from other modules of the system.

We initialize our feature set with a simplified contour of the person. This information is obtained from the background/foreground segmentation. The contour of the largest connected component is then approximated by line segments to simplify its shape.

As the facial expressions are the main part of the communication process, we also add them to ensure that they are correctly aligned. The facial features, we used are the eyes, nose, and mouth, and their positions are obtained specified currently by hand but eventually they will be detected from the tracking subsystem.

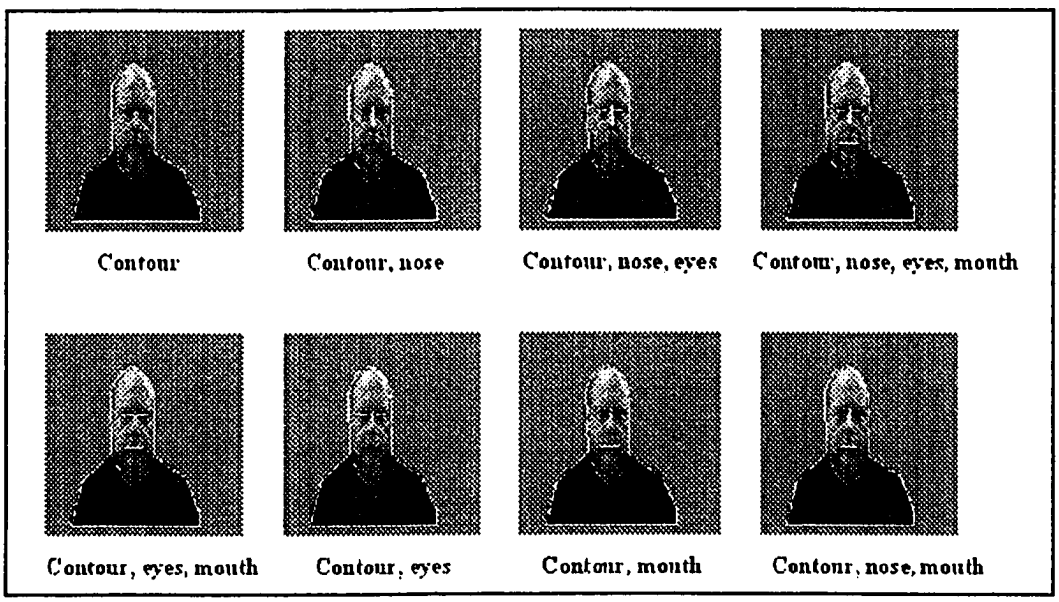


Figure 6.12: Input image with different feature sets.

Figure 6.12 shows the input image with the eight different feature sets selected for our experiments. For each feature set an intermediate view was generated and the result compared to a reference image with all the facial features present. Results obtained from this experiments are summarized in Table 6.1 for the stereo pairs of two different persons.

Feature set	Stereo pair 1	Stereo pair 2
Contour	1.28%	1.60 %
Contour, nose	0.47%	0.60 %
Contour, nose, eyes	0.31 %	0.18 %
Contour, nose, eyes, mouth	Reference	Reference
Contour, eyes, mouth	0.36 %	0.90 %
Contour, eyes	0.60 %	1.05 %
Contour, mouth	0.62 %	1.42 %
Contour, nose, mouth	0.22 %	0.43 %

Table 6-1: Results with different feature sets.

Different pairs of generated images present similar results with each of the feature sets used for their creation. From previous results, it is evident that images generated with only the contour have the poorest quality while the ones generated with either the eyes and nose or the mouth and nose present the best results.

Numeric results from Table 6.1 are helpful to give an idea of the quality of the generated images. However, it is from image observation that the best features can be selected. We noticed that the contour combined with either the nose and the eyes or, the nose and the mouth, give the best results. This is due to the fact that the nose performs a vertical alignment, while the eyes and mouth a horizontal alignment of the remaining facial features.

However, when the eyes are selected as a feature, they are the ones that correspond to the most visually convincing morph. One of the main reasons for this result, may reside in the natural tendency for people to look in the eyes of other people.

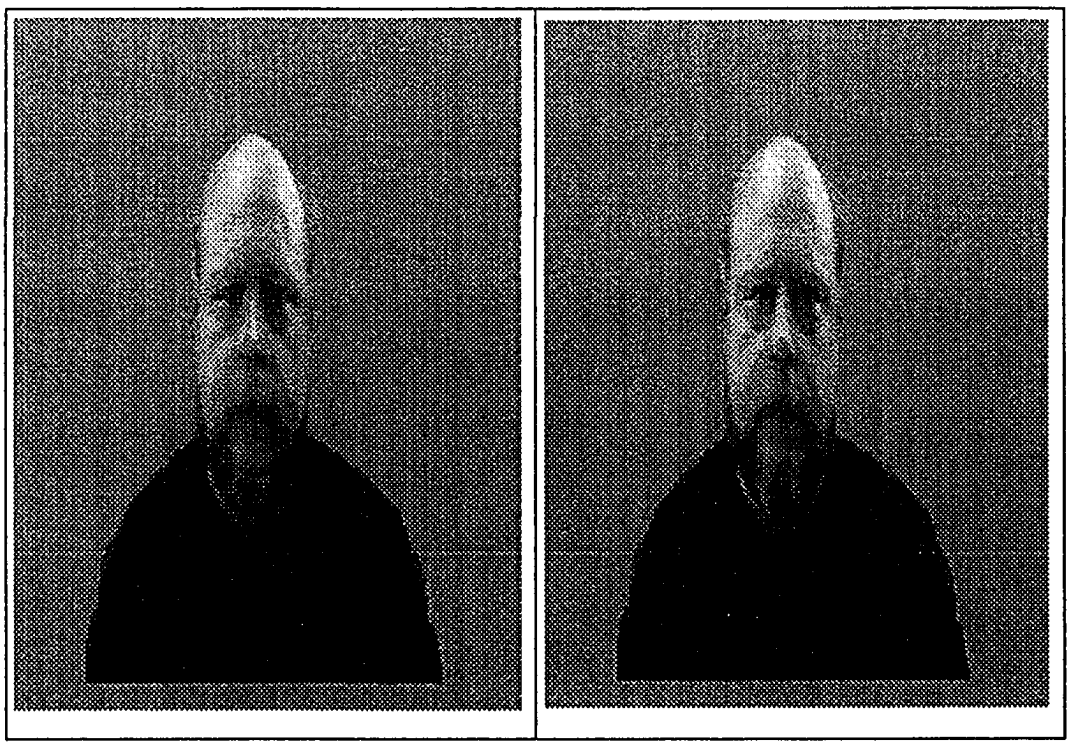


Figure 6.13: shows the view morph results for the worst (left) and best (right) feature sets.

6.3.4 View–morphing with Three Cameras

It is clear that view–morphing is able to generate life-like images of views found in–between a pair of cameras. But the main objective of the use of view–morphing in a tele–immersive system is to be able to generate views that correctly represent the user gaze, making the receiver of the images perceive eye contact. Achieving the correct gaze position it is not always possible with only two cameras, thus the need to study results obtained from a three-camera configuration.

Our last experiments consist in testing the results obtained while using view–morphing with three cameras. The importance of this test is to find out if after running the two view–morphing passes, required for a multiple camera setting, the quality of the generated images is preserved.

Although images for the tele–immersive system could not be tested, results with other images show that the blurring effect present in the generated views does not increase dramatically.

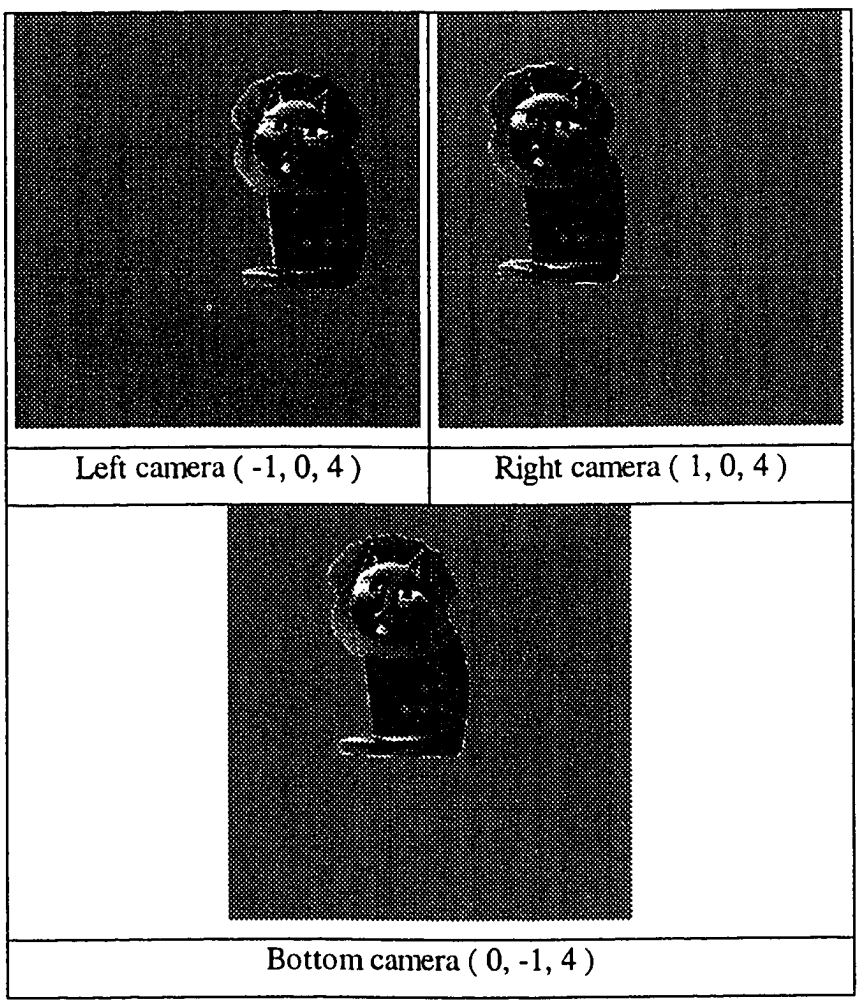
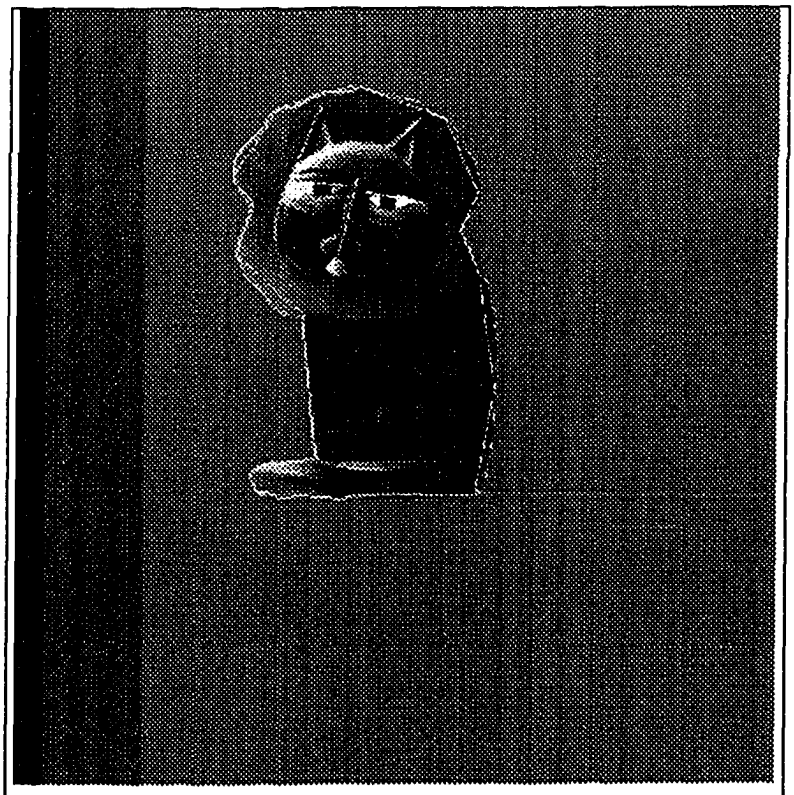


Figure 6.14: Input images for three-camera view–morphing.

In Figure 6.14, a set of three analyzed input images is shown. Following the same principle as with other images, we only used the contour and features on the cat's face for the morphing process. One of the resulting images, the one in the middle of the triangle formed by the three input images, is shown below with the feature set superimposed.



Camera position (0, -0.5, 4)

Figure 6.15: Resulting image of view morph using three cameras.

In the resulting image (6.15) 20.5% of the pixels present a different value from the one they have in the acquired image at the same position. The result is not bad considering that the differences in the coffin pair were of 16%, running only one pass of the view–morphing algorithm. This proves that the second pass does not degrade the quality of the image dramatically.

Another consideration to explain this result is that although the generated image and the real one acquired with a camera are supposed to be in the same position, some misalignments occurred that increased the number of pixels with differing values.

However, the result is encouraging because as it is seen from the image above, the face of the cat can be seen clearly since the nose and eye features preserve the correct alignment of the face. Where the most blurring can be perceived is in the body of the cat since strong features on it were ignored for the morph (e.g., circles on the body and leg edge).

6.4 Discussion

In this chapter we presented the different choices made for the implementation of both the hardware and software components of our prototype tele-immersive system.

The results presented demonstrate the suitability of using view-morphing in a tele-immersive system. View-morphing has many advantages over other view generation techniques, like the simplicity of its implementation, the possibility to adapt it to meet the needs of our system (real-time, automation) and the quality of the images produced.

Another benefit came from choosing feature-based image metamorphosis for the morph step. Given that its representation of image features is based on line segments, features can automatically be obtained from the images by the processing described in the previous sections (background segmentation, contour simplification and facial feature tracking).

In order to be able to integrate our view–morphing implementation to the tele–immersive system we still need to work on its time performance. Currently our results with sets of ten to 40 features, and images of 512 x 512 are obtained in approximately one second. However, the implementation does not take advantage of the scanline properties of the images and considers background pixels for the processing .

Chapter 7

Conclusions

In this thesis a prototype desktop tele–immersive system was proposed and some of its modules implemented and tested.

Tele–immersive systems were discussed in detail. The presentation started with the different alternatives to create and populate virtual environments. Next we reviewed some outstanding tele–immersive systems in order to define the context for our system.

From those discussions on tele–immersive systems, it can be seen that our system, although in its early implementation stages, has significant advantages. The use of an automated and real–time version of a view–morphing algorithm in the view generation module increases system performance. This method decreases the view synthesis time significantly while obtaining high quality images.

To improve the performance of the view generation module, all of its internal processes were carefully studied. We optimized the morphing process by restricting the number of edges used. To do this, we defined a minimum set of features needed for the morph of human faces. Based on the concept of the boundary flow studied by Seitz [102], we found that the contour of the person and facial features like the eyes and nose, provide enough information to the morph process. The number of tracked features depends on the complexity of the boundary of the stereo pair used for the system. However, we showed that with our implementation, simplified contours of the individuals worked very well and did not present significant loss of details.

As mentioned throughout this thesis, the implementation of all the modules of the system was beyond the scope of this work. In the near future, our work will be focused on the implementation of the missing modules of the system.

A first step will be the development of real–time eye and nose feature detectors. The use of robust infrared light techniques may be of use for this task. Having accurate position information for the eye and nose features would bring our system a step closer to its complete automation.

Our background segmentation module also has to be improved in order to achieve real–time operation. A first enhancement to this module is to include infrared information from the feature trackers. Position information of the facial features can be used as starting points to segment the user from the background applying a morphological growing technique as described in [118].

However, a good upgrade to the system will be the use of a Zcam [119]. Using this camera an accurate real–time segmentation of the user is obtained. Besides, depth information provided by the Zcam can also be used to the feature correspondence algorithm.

For the long–term, our idea is to take the principles of the system to a life–size meeting room. Although our desktop system provides some level of immersion, it does not provide the user with many natural communication elements (e.g. hand gestures, ability to move around the virtual world).

We plan to develop a tele–immersive system for the three–wall display system in the VizRoom [49]. A system of this size would increase the immersion experience by allowing the user to move around and interact with people face to face. In this setting the user will not have to adapt his/her behaviour in order to fit in the video window.

Although there are still many issues to be improved in order to make tele–immersive systems part of our everyday life, current systems show that they may represent a better communication alternative to current means of communication.

Bibliography

1. *AIM AOL Instant Messenger.*http://www.aol.ca/aim

2. *Dimension Technologies Inc.*http://www.dti3d.com/

3. *First Virtual Communications.*http://www.cuseeme.com

4. *MSN Messenger.*http://messenger.msn.com

5. *Shape Capture.*http://www.shapecapture.com

6. *TANDBERG Video Systems and Services.*http://www.tandberg.net

7. *Trillian.*http://www.trillian.cc/

8. *University of Alberta Access Grid Node.*http://www.ualberta.ca/CNS/RESEARCH/AccessGrid/

9. *VCON.*http://www.vcon.com

10. *VTEL.*http://www.vtel.com

11. *E. T. Surf Home: Mirabilis creates the first Internet wide Instant Messenger.* 1996: Tel-Aviv, Israel.http://company.icq.com/info/icqstory.html

12. Adelson, E.H. and J. Bergen, *The plenoptic function and the elements of early vision.* Computational Models of Visual Processing, 1991: p. 3-20.

13. Appel, A., *Some Techniques for Shading Machine Renderings of Solids.* SJCC, 1968: p. 37-45.

14. Avidan, S. and A. Shashua, *Novel view synthesis in tensor space.* IEEE Conference on Computer Vision and Pattern Recognition, 1997: p. 1034-1040.

15. Banerjee, P. and D. Zetu, *Virtual Manufacturing.* 2001: John Wiley.

16. Bao, P. and D. Xu. *Multiresolution image morphing in wavelet domain.* in *IEEE International Conference on Information Visualization.* 2000.

17. Barfield, W. and S. Weghorst, *The sense of presence within virtual environments: a conceptual framework.* Human-Computer Interaction: Software and Hardware interfaces, 1993.

18. Barnes, C. and E. Westbrook. *Remote Visualization of XRay Diffraction Data from the X8C Beamline at NSLS.* in *SIGGRAPH'92.* 1992. Chicago.

19. Barrus, J.W., R.C. Waters, and D.B. Anderson, *Locales and Beacons: Efficient and Precise Support For Large Multi-User Virtual Environments.* Proceedings of IEEE VRAIS, 1996.

20. Beier, T. and S. Neely. *Feature based image metamorphosis.* in *SIGGRAPH.* 1992.

21. Beier T., N.S. *Feature based image metamorphosis.* in *SIGGRAPH.* 1992.

22. Benford, S., et al., *Networked Virtual Reality and Cooperative Work.* Presence: Teleoperators and Virtual Environments, 1995. **4**(4): p. 364-386.

23. Boulanger, P., *Research Proposal.* 2001, University of Alberta: Edmonton

24. Boulanger, P. and M. Benitez. *A Tele-Immersive System for Collaborative Artistic Creation.* in *AVIR 2003.* 2003. Switzerland.

25. Brunelli, R. and T. Poggio, *Face Recognition: Features versus Templates.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1993. **15**(10): p. 1042-1052.

26. Burdea, G. *Virtual Reality Systems and Applications.* in *Electro'93 International Conference.* 1993. Edison,NJ.

27. Burl, M.C. and P. Perona. *Recognition of Planar Object Classes.* in *Computer Vision and Pattern Recognition.* 1996. San Francisco, CA.

28. Capin, T.K., et al. *Realistic Avatars and Autonomous Virtual Humans in: VLNET Networked Virtual Environments.* in *Virtual Worlds in the Internet.* 1998: IEEE Computer Society Press.

29. Carlsson, C. and O. Hagsand. *DIVE - a Multi-User Virtual Reality System.* in *Proceedings of IEEE VRAIS.* 1993. Seattle, Washington.

30. Chang, N.L. and A. Zakhor, *Arbitrary view generation for three-dimensional scenes from uncalibrated video cameras.* International Conference on Acoustics, Speech and Signal Processing, 1995. **4**: p. 2455-2458.

31. Chen, S.E. and L. Williams, *View interpolation for image synthesis.* Computer Graphics, 1993. **27**(Annual Conference Series): p. 279-288.

32. Chen, S.E. and L. Williams, *QuickTime VR - an image-based approach to virtual environment navigation.* Computer Graphics, 1995. **29**(Annual Conference Series): p. 29-38.

33. Cheung, G., S. Vedula, and T. Kanade, *Spanish Dancing (Flamenco) with music*.http://www-2.cs.cmu.edu/afs/cs/project/VirtualizedR/www/dance.html

34. Criminisi, A., et al., *i2i: 3D Visual Communication*, Microsoft Research i2i: Cambridge.http://research.microsoft.com/vision/cambridge/i2i/

35. Criminisi, A., et al. *Gaze Manipulation for One-to-one Teleconferencing.* in *ICCV'03*. 2003.

36. Davies, H., *GEA-01-013v2 (D5): Definition and Establishment of the European Distributed Access, v2.* 2001, GEANT. p. 17.http://www.dante.net/server/show/nav.007

37. Devebec, P.E. *Modeling and rendering architecture from photographs* Ph.D. Thesis Computer Science Division1996UC Berkeley154

38. Devebec, P.E., C.J. Taylor, and J. Malik. *FACADE: Modeling and Rendering Architecture from Photographs.* in *SIGGRAPH'96*. 1996.

39. El-Hakim, S. F., et al. *Two 3-D Sensors for Environment Modeling and Virtual Reality: Calibration and Multi-view Registration.* in *International Archives of Photogrammetry and Remote Sensing*. 1996. Vienna, Austria.

40. Faugeras, O., *Three-dimensional computer vision a geometric viewpoint.* Fourth edition ed. 2001, Cambridge, Mass.: Massachusetts Institute of Technology.

41. Faugeras, O.D. and S.J. Maybank, *Motion from point matches: multiplicity of solutions.* The International Journal of Computer Vision, 1990. **4**(3): p. 225-246.

42. Feldmann, S., et al. *Real-Time Segmentation for Advanced Disparity Estimation in Immersive Videoconference Applications.* in *10th International Conference on Computer Graphics, Visualization and Computer Vision.* 2002. Plzen, Czech Republic.

43. Frunze, A., *Echo Cancellation Demystified*, SPIRIT Corporation. p. 1-13.http://www.spiritcorp.com/pdf/article_4.pdf

44. Gibbs, S.J., C. Arapis, and C.J. Breiteneder, *TELEPORT - Towards immersive copresence.* Multimedia Systems, 1999. 7: p. 214-221.

45. Goral, C.M., et al. *Modeling the interaction of Light Between Diffuse Surfaces.* in *SIGGRAPH'84.*

46. Gorodnichy, D.O. *On Importance of Nose for Face Tracking.* in *International Conference on Automatic Face and Gesture Recognition (FG'2002).* 2002. Washington D. C.

47. Gorodnichy, D.O., S. Malik, and G. Roth, *Affordable 3D Face Tracking using Projective Vision.* Proceedings of International Conference on Vision Interface (VI'2002), 2002.

48. Gortler, S.J., et al., *The lumigraph.* Computer Graphics Proceedings, 1996. **Annual Conference Series**(SIGGRAPH'96): p. 43-54.

49. Green, M., *VizRoom User's Manual.* 1999, University of Alberta: Edmonton, AB. p. 7.http://www.cs.ualberta.ca/~graphics/MRToolkit/cave/user.pdf

50. Greenhalgh, C. and S. Benford. *MASSIVE, A Distributed Virtual Reality System Incorporating Spatial Trading.* in *15th International Conference on Distributed Computing Systems.* 1995. Los Alamitos, CA: ACM.

51. Harlyn Baker, H., et al., *Computation and Performance Issues in Coliseum, An Immersive Videoconferencing System.* 2003, HP Laboratories: Palo Alto Ca.

52. Harlyn Baker, H., et al., *The Coliseum Immersive Teleconferencing System.* 2002, HP Laboratories: Palo Alto Ca.

53. Hartley, R. and A. Zisserman, *Multiple view geometry in computer vision.* 2002, Cambridge: Cambridge University Press.

54. Holliman, N., *Auto-stereoscopic displays: Personal VR for the office and home.* 2001, Visualization Research Group University of Durham.http://www.dur.ac.uk/n.s.holliman/Presentations/DTI-2001-1up.pdf

55. Hsu, R., K. Kodama, and H. Harashima, *View interpolation using epipolar plane images.* IEEE International Conference on Image Processing, 1994. **2**: p. 745-749.

56. Huang, H.-C., C.-C. Kao, and Y.P. Hung, *Generation of multiviewpoint from stereoscopic video.* IEEE Transactions on Consumer Electronics, 1999. **45**(1): p. 124-134.

57. Insley, J.A., D.J. Sandin, and T.A. DeFanti, *Using Video to Create Avatars in Virtual Reality.* International Conference on Computer Graphics and Interactive Techniques, 1997: p. 128.

58. Izquierdo M., E. and M. Ghanbari, *Virtual 3D-View Generation from Stereoscopic Video Data.* IEEE International Conference on Systems, Man and Cybernetics, 1998. **2**: p. 1219-1224.

59. Ji, Q. and X. Yang, *Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance.* Real-Time Imaging, 2002. **8**: p. 357-377.

60. Kanade, T., H. Saito, and S. Vedula, *The 3d room: Digitizing time-varying 3d events by synchronized multiple video streams.* 1998, Robotics Institute Technical Report

61. Katkere, A., et al., *Towards video-based immersive environments.* Multimedia Systems, 1997. **5**(2): p. 69-85.

62. Klette, R., K. Schluns, and A. Koschan, *COMPUTER VISION Three-Dimensional Data from Images.* 1998, New York: Springer-Verlag.

63. Koenen, R., *MPEG-4 Overview.* 2002, International Organization for Standardization. p. 79.http://www.m4if.org/

64. Lanier, J., *Virtually there,* in *Scientific American.* 2001. p. 66-75

65. Laveau, S. and O. Faugeras, *3-d scene representation as a collection of images and fundamental matrices.* 1994, Institut national de recherche en informatique et automatique

66. Lavery, D., *The Future of Telerobotics.* Robotics World, 1996(Summer 1996).

67. Lengyel, J., *The Convergence of Graphics and Vision.* Computer, 1998: p. 46-53.

68. Leung, W.H., et al. *Realistic video avatar.* in *IEEE International Conference on Multimedia and Expo.* 2000. New York.

69. Levoy, M. and P. Hanrahan, *Light field rendering.* Computer Graphics Proceedings, 1996. **Annual Conference Series**(SIGGRAPH'96): p. 31-42.

70. Lhuillier, M. and L. Quan, *Image interpolation by joint view triangulation.* IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999. **2**: p. 145.

71. Lie, W.-N. and B.-E. Wei, *Intermediate view synthesis from binocular images for stereoscopic applications.* The 2001 IEEE International Symposium on Circuits and Systems, 2001. **5**: p. 287-290.

72. Liu, C., *A Bayesian Discriminating Features Method for Face Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003. **25**(6): p. 725-740.

73. Manning, R.A. and C.A. Dyer, *Interpolating view and scene motion by dynamic view morphing.* IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999. **1**: p. 394.

74. Mansouri, A.-R. and J. Konrad, *Bayesian winner-take-all reconstruction of intermediate views from stereoscopic images.* IEEE Transactions on Image Processing, 2000. **9**(10): p. 1710-1722.

75. Martin i Rull, E.X. and A.B. Martinez Velasco, *Generation of Synthetic Views for Teleoperation in Industrial Processes.* 2001.

76. Maybank, S.J. and O.D. Faugeras, *A Theory of Self-Calibration of a Moving Camera.* The International Journal of Computer Vision, 1992. **8**(2): p. 123-151.

77. McIvor, A., Q. Zang, and R. Klette, *The Background Subtraction Problem for Video Surveillance Systems.* 2000, Computer Science Department of The University of Auckland CITR at Tamaki Campus: Auckland, Australia. p. 1-13

78. McMillan, L. and G. Bishop, *Plenoptic modeling: An image-based rendering system.* Computer Graphics, 1995. **29**(Annual Conference Series): p. 39-46.

79. McNerney, P.J., J. Konrad, and M. Betke, *A Stereo-Based Approach to Digital Image Compositing.* IEEE Transactions on Multimedia, 2004(September 2004).

80. Milgram, P. and F. Kishino, *A Taxonomy of Mixed Reality Visual Displays.* IEICE Transactions on Information Systems, 1994. **E77**(D(12)): p. 1321-1329.

81. Moezzi, S., L.C. Tai, and G. P., *Virtual View Generation for 3D Digital Video.* IEEE Multimedia, 1997. **4**(1): p. 18-26.

82. Mulligan, J., et al., *Stereo-Based Scanning for Immersive Telepresence.* 2003

83. Nishita, T. and E. Nakamae. *Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection.* in *SIGGRAPH'85.* 1985.

84. O'Brien, P. and G. Watson, *Frequency Dependent Interpolation for Image Based Models.* The 20th Eurographics UK Conference, 2002: p. 75-82.

85. Ohm, J.-R., et al., *A real-time hardware system for stereoscopic videoconferencing with viewpoint adaptation.* Proceedings International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging, 1997.

86. Ohm, J.R., E. Izquierdo, and K. Muller, *Systems for disparity-based multiple-view interpolation.* IEEE International Symposium on Circuits and Systems, 1998.

87. Park, Y. and K. Yoon, *View Morphing using Sprites with Depth.* Fifth International Conference on Information Visualization, 2001: p. 323-328.

88. Phong, B.T., *Illumination for Computer Generated Pictures.* Communications of the ACM, 1975. **18**(6): p. 311-317.

89. Pollefey, M. *Tutorial on 3D Modeling from Images.* in *ECCV 2000.* 2000. Dublin, Ireland.

90. Pratt, D.R., et al., *Humans in Large-scale, Networked Virtual Environment.* Presence: Teleoperators and Virtual Environments, 1997. **6**(5): p. 547-564.

91. Radke, R., et al., *Using View Interpolation for Low bit-rate Video.* International Conference on Image Processing, 2001. **1**: p. 453-456.

92. Rajan, V., et al., *A Realistic Video Avatar System for Networked Virtual Environments.* Immersive Projection Technology Symposium, 2002.

93. Raskar, R., et al. *The office of the future: a unified approach to image-based modeling and spatially immersive displays.* in *SIGGRAPH.* 1998.

94. Rawlings, M., et al. *Using Virtual Reality for Machine Design.* in *SIGGRAPH'94.* 1994.

95. Riva, G. and J.A. Waterworth, *Presence and the Self: A cognitive neuroscience approach.* Presence-Connect, 2003. **3**(1).

96. Sadagic, A., et al. *National Tele-Immersion Initiative: Towards Compelling Tele-Immersive Collaborative Environments.* in *Medicine meets Virtual Reality.* 2001. Newport Beach, California.

97. Saito, H., et al. *Appearance-based virtual view generation of temporally-varying events from multi-camera images in the 3D Room.* in *Proceedings of Second International Conference on 3-D Digital Imaging and Modeling.* 1999. Ottawa, Canada.

98. Sandin, D., et al., *Video avatars in collaborative virtual environment.* 2002, Electronic Visualization Laboratory: Chicago.http://www.ice.eecs.uic.edu/research/res_project.php3?indi=165

99. Scharstein, D. *Stereo Vision for View Synthesis.* in *Computer Vision and Pattern Recognition (CVPR'96).* 1996. San Francisco, CA: IEEE.

100. Scharstein, D. and R. Szeliski, *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.* IJCV, 2002. **47**: p. 7-42.

101. Schuemie, M.J., et al., *Research on Presence in Virtual Reality: A Survey.* CyberPsychology & Behavior, 2001. **4**(2): p. 183-201.

102. Seitz, S. *Image-based transformation of viewpoint and scene appearance.* Ph. D. Thesis1997University of Wisconsin

103. Seitz, S. and C. Dyer. *View morphing.* in *SIGGRAPH.* 1996.

104. Shum, H.-Y., S.B. Kang, and S.-C. Chan, *Survey of Image-Based Representations and Compression Techniques.* IEEE Transactions on Circuits and Systems for Video Technology, 2003. **13**(11): p. 1020-1037.

105. Soucy, M., et al., *Sensors and algorithms for the reconstruction of digital 3-D colour models of real objects.* International Conference on Image Processing, 1996. **1**: p. 409-412.

106. St. Arnaud, B., *Proposed CA*net 4 Network Design and Research Program.* 2002. p. 47.http://www.canarie.ca/canet4/

107. Szeliski, R., *A multi-view approach to motion and stereo.* IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999. **1**: p. 163.

108. Talmi, K. and J. Liu, *Eye and Gaze Tracking for Visually Controlled Interactive Stereoscopic Displays.* Signal Processing: Image Communication, 1999. **14**: p. 799-810.

109. Theis, C. and K. Hustadt. *Detecting the gaze direction for a man machine interface.* in *IEEE Int. Workshop on Robot and Human Interactive Communication ROMAN 2002.* 2002. Berlin, Germany.

110. Thom, G.A., *H.323: the multimedia communications standard for local area networks.* IEEE Communications Magazine, 1996. **34**(12): p. 52-56.

111. Toyama, K., J. Krumm, and B. Meyers. *Wallflower: Principles of Background Maintenance.* in *International Conference on Computer Vision.* 1999. Corfu, Greece.

112. Visualization, W.C., *Building an Access Grid Station*, University of Alberta: Edmonton, AB. p. 19.http://www.ualberta.ca/CNS/RESEARCH/AccessGrid/StatusReports.html

113. Wang, H. and P. Chu, *Voice source localization for automatic camera pointing system in videoconferencing.* IEEE International Conference on Acoustics, Speech and Signal Processing, 1997. **1**: p. 187-190.

114. Wolberg, G., *Image morphing: a survey.* The Visual Computer, 1998. **14**: p. 360-372.

115. Wolberg, G., et al., *One dimensional resampling with inverse and forward mapping functions.* Journal of Graphics Tools, 2000. **5**(3): p. 11-33.

116. Wong, V., *Telepresence in Medicine: An Application of Virtual Reality.* SURPRISE 96 Journal, 1996. **2**.

117. Wong, W. *View interpolation for shared virtual environments.* Master Thesis Faculty of Administration. 2004 University of Ottawa p.103

118. Wren, C., et al., *Pfinder: Real-Time Tracking of the Human Body.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997. **19**(7): p. 780-785.

119. Yahav, G. and G. Iddan, *3DV Systems' Zcam.* Broadcast Engineering, 2002(July 2002).

120. Yang, Y., X. Wang, and J.X. Chen, *Rendering avatars in virtual reality: Integrating 3d model with 2d images.* IEEE Computing in Science and Engineering, 2002. **4**(1): p. 86-91.

121. Yeung, F., *Internet 2: scaling up the backbone for R&D.* IEEE Internet Computing, 1997. **1**(2): p. 36-37.

122. Zeltzer, D., *Autonomy, interaction, and presence.* Presence: Teleoperators and Virtual Environments, 1992. **1**(1): p. 127-132.

123. Zhang, L., D. Wang, and A. Vincent, *An Adaptive Object-based Reconstruction of Intermediate Views from Stereoscopic Images.* International Conference on Image Processing, 2001. **3**: p. 923-926.

124. Zhu, Y. and K. Fujimura. *3D Head Pose Estimation with Optical Flow and Depth Constraints.* in *3-D Digital Imaging and Modeling.* 2003.

125. Zhu, Z., K. Fujimura, and Q. Ji. *Real-Time Eye Detection and Tracking Under Various Light Conditions.* in *2002 ACM SIGCHI Symposium on Eye Tracking Research & Applications.* 2002. New Orleans, LA.