Haptics in Augmented Reality

James Vallino
Department of Computer Science
Rochester Institute of Technology
jrv@cs.rit.edu

Christopher Brown
Department of Computer Science
University of Rochester
brown@cs.rochester.edu

Abstract

An augmented reality system merges synthetic sensory information into a user's perception of a three-dimensional environment. An important performance goal for an augmented reality system is that the user perceives a single seamless environment. In most augmented reality systems the user views a real world augmented only with visual information and is not provided with a means to interact with the virtual objects. In this paper we describe an augmented reality system that, in addition to visual augmentation, merges synthetic haptic input into the user's perception of the real environment. Our system uses a ***PHANToM***™ *haptic interface device to generate the haptic sensory input in real-time. The system allows user interactions such as moving or lifting a virtual object, and demonstrates interactions between virtual and real objects. Methods to provide proper visual occlusion between real and virtual objects are also described.*

1. The Challenge of Augmented Reality

There has been considerable interest in augmented reality (AR) systems that mix live video from a camera with computer-generated objects registered in a user's three-dimensional environment [1]. Applications of this powerful visualization technique include maintenance tasks [2], surgical planning [3, 4], and new user interfaces [5]. The resulting AR systems allow three-dimensional virtual objects to be visually embedded into a user's perception of the environment.

In a typical AR system, Figure 1, a video camera views the real scene and generates a 2D image of it on the image plane. The user sees an augmented view composed of a synthetic graphics image merged with the image of the real scene. To maintain the illusion that the virtual objects are indeed part of the real world requires a consistent registration of these two images—the major challenge for augmented reality systems.

This registration requirement for creating a high-fidelity augmented reality environment can be stated in terms of the relationships that must be determined and maintained (Figure 1). The object-to-world transform **O**, specifies the position and orientation of a virtual object with respect to the world coordinate system that defines the real scene. The world-to-camera transform **C**, defines the location and orientation of the video camera that views the real scene. Finally, the camera-to-image plane transform **P**, specifies the projection operation the camera performs to create a 2D image of the 3D real scene. Any errors in the determination of these relationships appear to the user as inconsistencies in the appearance and position of the virtual objects in the real scene.

To faithfully create haptic interactions with the virtual objects there is also a registration problem between the real world and the system generating the haptic display. There is a haptic-to-world transform that defines the relationship between the world coordinate system and the coordinate system in which the haptic interface operates. Accurately computing these relationships while maintaining real-time response and a low latency is the primary performance goal for a haptically and visually augmented reality system.

2. Augmenting Reality Using Affine Representations

Only when the relationships between the multiple coordinate systems shown in Figure 1 are known can the synthetic sensory information correctly merge into the user's perception of the real scene. Traditional AR systems approach the problem of computing these transforms using sensing, calibration and measurement to explicitly determine each transform [6]. These systems use sensors to measure the camera's pose with respect to the world coordinate system thus determining the world-to-camera transform, **C**. Quantifying the

The support of the National Science Foundation under Grant No. CDA-94-01142 and the Defense Advance Research Projects Agency under Grant No. DAAB07-97-C-J027 is gratefully acknowledged.

0-7695-0253-9/99 $10.00 © 1999 IEEE

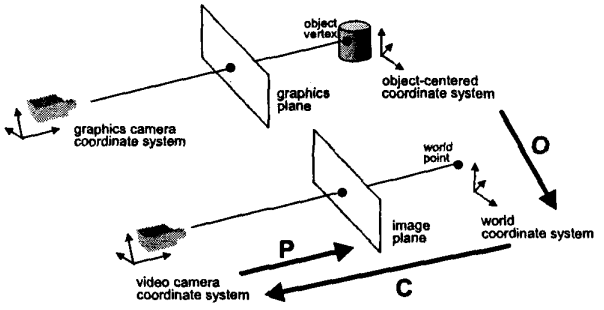


Figure 1 - Augmented reality coordinate systems

camera-to-image transform, **P**, requires knowledge of the camera's intrinsic parameters [7]. The third transform, **O**, is computed by measuring the desired position for the virtual object in the world coordinate system. From this all the necessary transforms are known so that, at least in principle, virtual objects can be rendered and merged correctly with the live video.

The methods based on position measurements exhibit errors due to inaccuracies and latencies in position sensing, and errors in the camera calibration parameters. A novel aspect of our augmented reality system (not pursued here) is that it requires no a priori metric information about the intrinsic and extrinsic parameters of the camera, where the user is located in the world, or the position or geometry of objects in the world [8]. In our system we track four features in real-time and define the global affine coordinate system solely from the location of those tracked features in the video image. All relationships are determined in this common affine coordinate system

3. An Haptic Augmented Reality Interface

None of this prior work has included any interaction with the virtual objects except for the visual changes in the augmented reality display whenever the user changes viewpoint. One of the reasons stated by Mine, Brooks, et. al. [9] for the paucity of virtual-environment applications that have left the laboratory setting is the lack of haptic feedback. Previous haptic research is concentrated in the areas of telemanipulation and virtual reality. The work in these areas does not, however, provide insights into the problems of registration with the real scene or interactions between real and virtual objects.

3.1. Haptic technology

Haptic technology provides the user with an ability to experience touch sensations. We emphasize user interaction with a natural-seeming augmented environment requiring the user to have a sense of feeling the virtual objects, touching their surface, and interacting with them in a dynamic fashion. Some work in virtual reality applications demonstrated haptic interfaces. In Project GROPE at the University of North Carolina [10], the user explores molecular docking by manipulating molecules in an immersive virtual environment using a large-scale force-reflexive manipulator. Ziegler, Brandt, et. al. [11] describe a simulator for arthroscopic surgery that uses force feedback in the virtual environment to train surgeons in the procedure. Using the Rutgers Master II force feedback device Dinsmore, Langrana, et. al. [12] built a virtual environment for training physicians to locate and palpate tumor masses.

The work of State, Hirota et. al. [13] shows interaction with virtual objects. There is neither haptic feedback nor dynamic interactions between the virtual and real objects however. Yokokohji, Hollis et. al. [14] demonstrate a haptic interface for an AR system with neither motion of virtual objects nor interaction between virtual and real objects in their system.

To allow realistic interactions with the virtual objects we chose the **PHANToM™** [15]. This device looks similar to a small robot arm with a motor driving each joint. The controller drives the motors to generate the requested force feedback at the end effector, a thimble into which the user inserts a finger. With the supplied GHOST™ library the system defines a world of objects in a haptic scene to be "rendered". Mass can be assigned to the objects so when the user places a finger under an object and lifts, the weight of the object is felt resting on the finger. It is also possible to simulate object surface compliance and texture. Figure 2 shows the system diagram for our haptic AR system.

3.2. Haptic-graphic interaction

In Section 2 we described the method that we use for registering the world and graphics coordinate systems. We accomplish this registration by defining a common

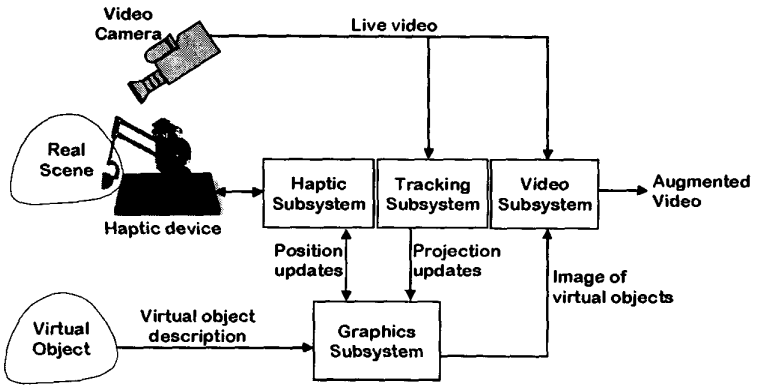


Figure 2 - Components of Augmented Reality System

global affine coordinate system. The system creates a scene graph containing the virtual objects defined in an Euclidean coordinate system. We compute a transform from this coordinate system to the global affine coordinate system as part of the process to place virtual objects in our real scene. The GHOST library allows definition of a haptic scene in a manner analogous to defining the graphic scene used for generating the virtual objects.

Our haptic AR system must establish the relationship between the Phantom's haptic coordinates and the global affine coordinates in which all of the virtual graphic objects are described. Once we establish that relationship it is possible to exchange position information between the haptic and graphic scene so that our system can instruct the Phantom to generate haptic feedback to the user appropriate for the interactions with the virtual objects.

Our system computes the Phantom to affine transform, $\mathbf{T}_{ap}$, by having the user move the Phantom end effector to four points in the workspace for which affine coordinates are known. The system records the position in the Phantom coordinate system at each point. $\mathbf{T}_{ap}$ is computed by solving:

$$[\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3 \quad \mathbf{a}_4] = \mathbf{T}_{ap}[\mathbf{p}_1 \quad \mathbf{p}_2 \quad \mathbf{p}_3 \quad \mathbf{p}_4]$$

where $\mathbf{a}_i$ and $\mathbf{p}_i$ are, respectively, the affine coordinates of the points in the workspace and their corresponding haptic coordinates.

A tight coupling exists between the graphics scene and haptic scene as Figure 3 depicts. At each cycle of operation the system obtains the current Phantom position and transforms it into the global affine coordinate system. This affine point is then transformed into the Euclidean coordinate system of the virtual object using an inverse of $\mathbf{T}_{ap}$. The location of this point on the virtual object determines the appropriate haptic response. The system sends the correct commands to the Phantom to generate the feedback at the user's fingertip, so textures that look different also feel different.

3.3. Virtual-to-virtual haptic interaction

The system allows several different interactions with virtual objects. (MPEG video clips of all the demonstrations described in this paper can be found at http://www.cs.rit.edu/~jrv/research/ar/.) In a typical demonstration, a virtual globe appears in the augmented image. When the globe is stationary (Figure 4) the user

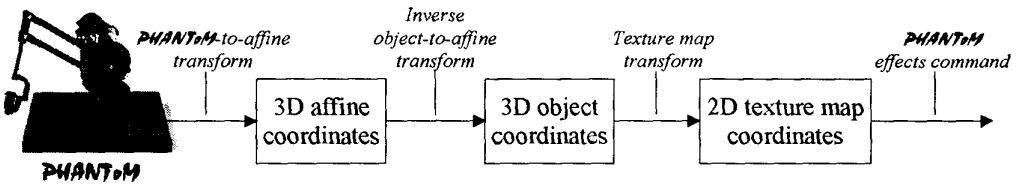


Figure 3 - Phantom-graphics coupling

can move a finger over the surface and receive correctly registered haptic sensations to help distinguish water and land masses. Another demonstration spins the globe on its axis. The system tracks the location of the active point of the Phantom on the globe surface so that haptic feedback changes in response to both user motions and the spinning of the globe.

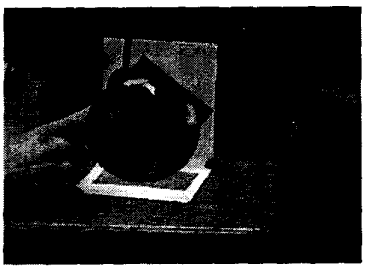


Figure 4 - Haptic interactions with a globe

3.4. Virtual-to-real haptic interactions

The next demonstration is with a cube virtual object. In Figure 6a, the user is moving the cube around by lifting it with a finger. The vertical part of the frame is defined in the haptic scene so that haptic interaction takes place between real and virtual objects. In Figure 6b, the user has just rested the virtual cube on top of the real vertical wall. It remains suspended in that position until moved by the user.

In an augmented view of the scene visual interactions between real and virtual objects must be considered [8]. The virtual camera in the computer graphics correctly handles hidden surface elimination within a virtual object and between virtual objects. The visual interaction between real and virtual objects must be considered. Using the graphics as the foreground element, or key mask, a luminance keyer displays the graphics image at every pixel above the luminance key value. The live video image is shown whenever a graphics pixel has a luminance value less than the key value. If the background regions of the virtual graphics image are rendered with a luminance at or below the key value then the live video is shown in all background regions and the virtual objects will occlude the video image in the areas where a virtual object is rendered.

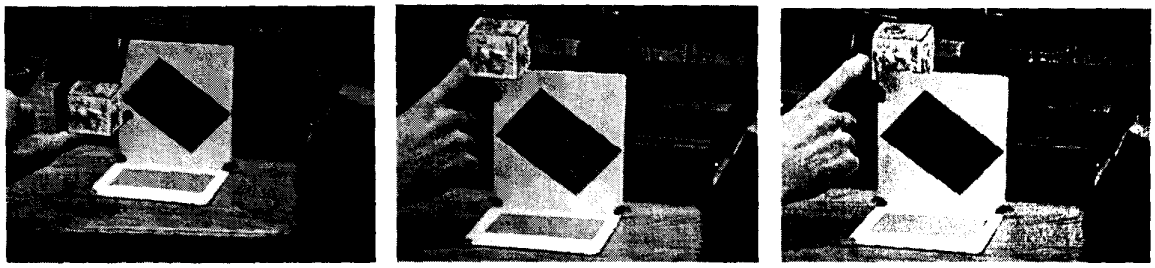
Hidden surface removal does not occur when a real object occludes a virtual one because there is no information about the geometric relationship between these objects [16]. If an affine model of a real object is included as another virtual object and rendered in the background color the keyer correctly resolves the occlusions. In our demonstrations, the vertical part of the frame is defined in the graphics scene as another virtual object. Figure 6c shows the cube disappearing behind the vertical wall after being pushed slightly by the user.

3.5. Foreground detection for improved visual occlusion

The images in Figure 4 and Figure 6 show one strong limitation of haptic interactions in this augmented reality system. This limitation is that the Phantom provides a very compelling sense of touching virtual objects but the visual image in the augmented display is not as compelling. The reason for this is that proper visual occlusions are not occurring between the virtual objects and the user's hand. In these examples, the virtual objects always occlude the user's hand even when the hand is interacting with a front surface of the object. A technique to ameliorate this problem is to create a red marker, defined as an object in the global affine coordinate system, representing the active point of the Phantom end effector. It facilitates the user's interaction with the virtual objects by being a substitute for the cues that visual occlusion normally provides.

Section 3.4 describes the method used for proper occlusion between real and virtual objects. To apply that idea to the haptic subsystem requires the Phantom device to be defined as a graphic object with actual joint angles monitored during operation controlling its configuration. However, this still does not provide for occlusion of virtual objects by the user's finger and hand.

A more general approach is to use a technique for foreground detection that has previously been applied to detecting humans moving through rooms. [17]. The technique initially analyzes the "background" workspace scene over a specified number of video frames and computes mean and covariance statistics on the YUV values of the video signal. The assumption is that the real scene contains objects that will always be present while the system is operating. When the system is running, any YUV pixel value found to be statistically different than the background color is marked as a foreground block. The foreground regions identify areas of the image where occlusion by a real object should take place. The system uses a 90% confidence level for the statistical test. Two methods have been implemented. The first gathers mean and covariance statistics on all three individual components in the YUV color space. A second method, which is more robust to shadow effects, computes statistics on only the UV chrominance components normalized by the luminance Y. All statistically different regions are deemed to represent real



(a) (b) (c)

Figure 6 - Interactions with a virtual cube

objects that have entered the workspace at the depth of the active point of the Phantom end effector.

To get the proper visual interactions we texture map the foreground plane with the detected foreground information. Figure 5a shows the foreground plane with detected areas of foreground activity marked in the regions with projections of virtual graphic objects. These are the only regions of interest. We want the virtual graphics image to key the live video into the final augmented image in any foreground region where the foreground mask plane is the virtual object closest to the camera in affine space. In the texture map applied to the foreground plane we render a detected foreground block as an opaque block below the luminance key value. In areas where we are not doing foreground detection or have detected no foreground activity we need the foreground plane to have no effect on the final virtual image or be transparent. The augmented image (Figure 5b) shows the user's finger properly occluding the virtual object that it hides.

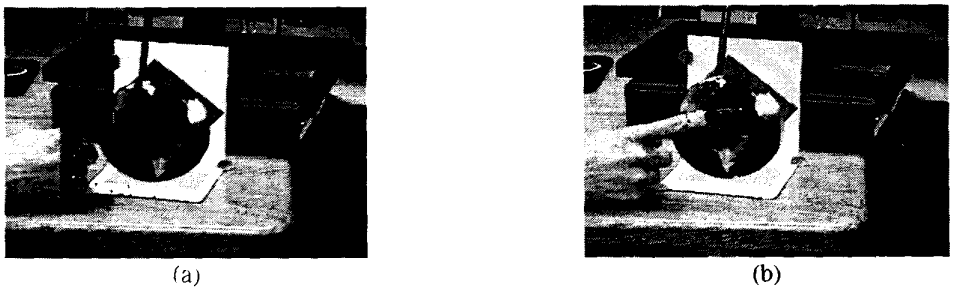
4. Discussion

The Phantom provides a very compelling sense of touching virtual objects especially when there is no conflict with the visual cues. Color background detection and masking of the video is a promising way to eliminate conflicts when proper occlusions between the real Phantom and the virtual objects do not occur.

Section 2 gave a brief overview of the uncalibrated, relative coordinate technique our system uses for registering the coordinate systems. This method is an attractive alternative to methods that require position measurement and/or camera calibration. It has limitations for the visual augmentation though [8]. There are implications on the haptic side also. As mentioned in Section 2 the global affine coordinate system is a non-Euclidean coordinate system. The technique we adopted computes the dynamics in the Euclidean reference frame of the Phantom and then transforms the haptic objects into the global affine coordinate system to determine the position for rendering their visual counterparts. Whether the dynamics can be computed directly in the global affine coordinate system is a question for future research.

5. Conclusions

Since its inception computer graphics has been interactive in nature. Augmented reality systems have been interactive only to the extent that the user could move about the workspace and be a passive viewer of the visually augmented scene. We have implemented an augmented reality system that incorporates a real-time



(a) (b)

Figure 5 - Foreground detection for visual occlusion

haptic interface device, thus adding touch as a second modality of synthetic sensory information augmenting the user's perception of a real scene. The user can realistically interact with a virtual object. These interactions include feeling the surface of the object, feeling the weight and dynamic forces of the object and moving the object within the workspace in a variety of manners. Future work aimed at decreasing system latency, better handling occlusions by real objects and scaling up the system will improve the performance of this augmented reality interface.

6. References

[1] R. T. Azuma, "A Survey of Augmented Reality," *Presence*, vol. 6, pp. 355-385, 1997.

[2] T. P. Caudell and D. Mizell, "Augmented reality: An application of heads-up display technology to manual manufacturing processes," presented at Proc. Hawaii Int. Conf. System Sciences, 1992.

[3] W. E. L. Grimson, G. J. Ettinger, S. J. White, P. L. Gleason, T. Lozano-Perez, W. M. W. III, and R. Kikinis, "Evaluating and Validating an Automated Registration System for Enhanced Reality Visualization in Surgery," presented at Proceedings of Computer Vision, Virtual Reality, and Robotics in Medicine '95, Nice, France, 1995.

[4] A. State, M. A. Livingston, W. F. Garrett, G. Hirota, M. C. Whitton, E. D. Pisano, and H. Fuchs, "Technologies for augmented reality systems: Realizing ultrasound-guided needle biopsies," presented at Proc. SIGGRAPH '96, 1996.

[5] P. Wellner, "Interacting with paper on the digitaldesk," *Comm. of the ACM*, vol. 36, pp. 86-95, 1993.

[6] M. Tuceryan, D. S. Greer, R. T. Whitaker, D. E. Breen, C. Crampton, E. Rose, and K. H. Ahlers, "Calibration Requirements and Procedures for a Monitor-Based Augmented Reality System," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, pp. 255-273, 1995.

[7] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Transactions of Robotics and Automation*, vol. RA-3, pp. 323-344, 1987.

[8] K. N. Kutulakos and J. R. Vallino, "Calibration-Free Augmented Reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, pp. 1-20, 1998.

[9] M. R. Mine, F. P. Brooks, and C. H. Sequin, "Moving objects in space: exploiting proprioception in virtual-environment interaction," presented at Proceedings of 24th International Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, 1997.

[10] F. P. Brooks, M. Ouh-Young, J. J. Batter, and P. J. Kilpatrick, "Project GROPE - Haptic Displays for Scientific Visualization," *Computer Graphics*, vol. 24, pp. 177-185, 1990.

[11] R. Ziegler, C. Brandt, C. Kunstmann, W. Müller, and H. Werkhäuser, "Haptic Display of the VR Arthroscopy Training Simulator," presented at Proceedings SPIE Vol. 3012: Stereoscopic Displays and Virtual Reality Systems IV, San Jose, CA, USA, 1997.

[12] M. Dinsmore, N. Langrana, G. Burdea, and J. Ladeji, "Virtual Reality Training Simulation for Palpation of Subsurface Tumors," presented at Proceedings of the IEEE 1997 Virtual Reality Annual International Symposium, Albuquerque, NM, USA, 1997.

[13] A. State, G. Hirota, D. T. Chen, W. F. Garrett, and M. A. Livingston, "Superior augmented reality registration by integrating landmark tracking and magnetic tracking," presented at Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, 1996.

[14] Y. Yokokohji, R. L. Hollis, and T. Kanade, ""What You can See Is What You can Feel" - Development of a Visual/Haptic Interface to Virtual Environments," presented at Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium, Santa Clara, CA, USA, 1996.

[15] J. K. Salisbury and M. A. Srinivasan, "Phantom-based haptic interaction with virtual objects," *IEEE Computer Graphics and Applications*, vol. 17, pp. 6-10, 1997.

[16] M. M. Wloka and B. G. Anderson, "Resolving Occlusion in Augmented Reality," presented at Proceedings 1995 Symposium on Interactive 3D Graphics, Monterey, 1995.

[17] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780-785, 1997.

Virtual $3D$ Interactions between $2D$ Real Multi-Views

Katia Fintzel Jean-Luc Dugelay

Institut Eurécom, Multimedia Communications Department
B.P. 193, 06904 Sophia Antipolis Cedex, France
E-mail: {fintzel, dugelay}@eurecom.fr

Abstract

In this paper, we introduce an image processing tool, Video Spatialization, as a new approach to navigate through a "virtualized" scene, only known by a limited set of $2D$ uncalibrated images (i.e without any $3D$ CAD model). We develop this approach in the context of an interactive application: a multipoint teleconferencing system for very low bit rate links (internet, mobile communications), based on the immersion of the $3D$ virtual models of all the participants, in a common virtualized meeting place controlled by video spatialization. This article contains (i) the recall of an efficient "mesh-oriented" algorithm for the reconstruction of real views and the synthesis of virtual ones from a triplet of uncalibrated views; (ii) some extensions of the original approach from one to n triplets of views to simulate $3D$ navigation and (iii) preliminary investigations using video spatialization for background control within the context of our virtual teleconferencing application.

1. Introduction

This paper discusses the problem of reconstructing real points of view of an arbitrary $3D$ scene and synthesizing virtualized ones, in order to simulate $3D$ navigation from a limited set of $2D$ uncalibrated views without resorting to any $3D$ CAD model of the scene: this is referred to as *Video Spatialization*. This technique aims at offering the possibility for an observer to visualize a scene from anywhere and in any direction, just as in real navigation through a $3D$ place. With this respect, we recall in section 2 an efficient "mesh-oriented" approach for real view regeneration from two neighboring ones, and a set of analytical inferences to synthesize the virtual point of view in relation with the user's motion and orientation (as developped in [10]). Image mosaïcking is therefore used as an image processing approximation to increase the overlapping areas between the initial input data ($2D$ uncalibrated images), and improve the visual rendering and realism of the resynthesized point of view. These just mentionned algorithms, called "intra-triplets" processings in this paper, deal only with three views of a real scene. However, applications for immersive media will use in practice more than three views. In section 3, we extend the "intra-triplets" process combining the synthesis method from several triplets of uncalibrated images, with image mosaïcking approximations applied to virtual output views, in order to simulate the larger real motions of an observer in a "virtualized" $3D$ scene. Such extensions are called "inter-triplets" processings. Finally, we describe in section 4 our future investigations in the context of the TRAIVI project, which takes advantage of video spatialization techniques for virtual teleconferencing systems, in which we introduce $3D$ clones inside a virtualized $2D$ meeting room.

2. Image Transfer

2.1. A "Mesh-Oriented" Approach

By extension of the stereovision concepts [8, 7], we proposed in [4, 10] an algorithm for real view reconstruction from uncalibrated $2D$ views of a $3D$ scene. This was based on trilinear tensors, first modeled by Spetsakis and Aloimonos [19] in the calibrated case and by Shashua in the uncalibrated case [16]. An existing view can therefore be reconstructed from two other neighboring views without any explicit calibration stage as follows:

- Analysis: Using seven or more corresponding points in three original uncalibrated views, eighteen parameters of a trilinear form can be estimated (for more details about the definition of trilinear parameters see [16, 17, 18] and [4]).
- Synthesis: The central view is reconstructed using all corresponding points of the external images (i.e left and right) and the estimated parameters from the analysis, as shown in figure 1.

0-7695-0253-9/99 $10.00 © 1999 IEEE

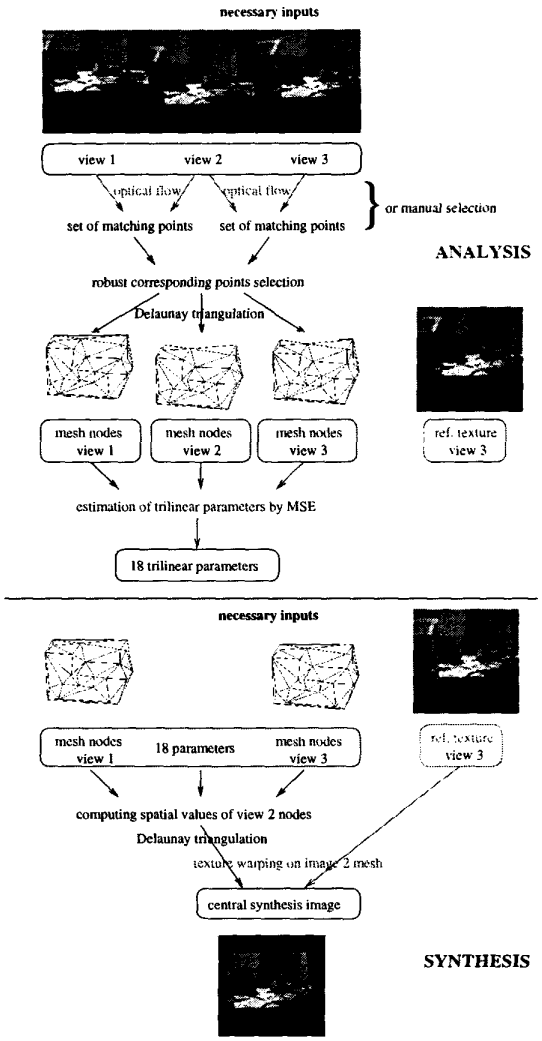
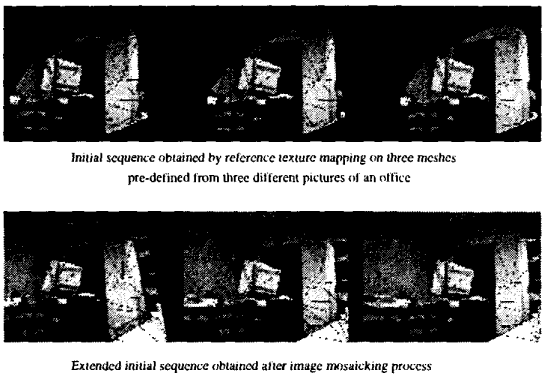


Figure 1. Regeneration process of a real view

Contrary to the image-based rendering methods for view synthesis typically found in the literature [1, 17, 18, 2] and resorting to dense correspondences, one of our contributions is to use a "mesh-oriented" approach. We represent the three original images as a reference texture, mapped on three associated meshes, defined using the Delaunay triangulation [21] on homologous points from the three initial images. This choice of representation is fully justified by the compatibility with real-time constraints of our applications and the increase of the visual comfort rather than the reconstruction accuracy, as explained in [10, 5]. As opposed to classical methods, we obtain plain reconstructions (without any mis/non or over-informed point) visually acceptable for our kind of applications. The coverage of the reconstruction obviously depends on the size of the common area of the three initial meshes, which can be very limited. As a solution to this problem, we introduce in the next subsection a module of image mosaïcking as a pre-processing step of the reconstruction method.

2.2. Mosaïcking on Original Triplets of Pictures

The initial image triplet used for the reconstruction method is represented by three meshes limited to a common covering area and a reference texture corresponding to the third image in our case. Using the theory of homographic transforms [6], we extend the initial meshes to the entire reference texture by approximations (as shown in figure 2).



Initial sequence obtained by reference texture mapping on three meshes pre-defined from three different pictures of an office

Extended initial sequence obtained after image mosaicking process

Figure 2. Initial and extended triplets

We hence obtain larger meshes, which may cover an area of the original image that was not investigated when initializing the meshes. By using image mosaïcking, we combine all the texture information of the initial data in the reconstruction.

2.3. Unknown Views Synthesis

The vector of trilinear parameters is analytically altered to simulate a virtual change of the focal length or a geometrical $3D$ displacement of the camera relative to the reconstructed view and synthesize unknown virtual points of view. Only some synthesis steps are required to simulate human relative motions or relative changes of the current point of view, whereas the analysis step remains unchanged. New views can therefore be virtualized in real time from well-adapted and relevant meshes, which depend on the $3D$ scene complexity and the desired quality for the synthesized images.

All the possible manipulations of the trilinear parameters and the inputs needed to simulate each kind of transformations, due to a change of the intrinsic or extrinsic parameters of the central video camera are summerized in [10]. And the complete developments concerning the modifications of the trilinear parameters, which render virtualized

consistent points of view of the scene, are reported in [4] for the simulations of the focal length changes and in [9] for all kinds of rotations and translations. Rotations and particularly translations are not straightforward without an explicit calibration, but our work aims at keeping the calibration step implicit by restoring the inputs (especially the relative rotations between the initial positions of the video cameras) directly from the trilinear parameters estimated from the triplet of initial images as explained in [9, 15].

To test the validity of the trilinear parameters estimation and manipulation, our method was first applied on several triplets of views extracted from a synthetic scene, before generating virtualized points of view from real scenes as shown in figure 3.

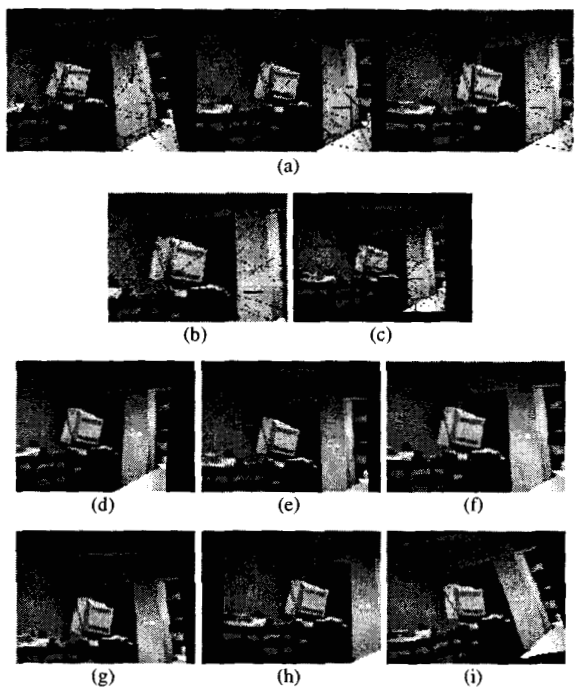


Figure 3. Synthesized points of view from extended meshes:
(a) initial triplet, (b,c) central video camera focal change, (d) central video camera translation along the horizontal axis, (e) along the vertical axis, (f) along the optical axis and (g) rotation around the horizontal axis , (h) around the vertical axis and (i) around the optical axis

3. Video Spatialization from Multi-Triplets

In order to simulate navigation through a virtualized scene, we must generate enough different synthesized images of the environment from several triplets of initial views, and link these resulting syntheses together as if the observer moves freely in the scene. The user's eyes are then considered as a virtual camera, whose positions and motions allow us to synthesize continually his coherent visual feedback of the scene.

Let us consider several triplets of images, represented by the needed reference textures and their associated meshes (a texture per triplet of meshes). We are able to simulate motions around each triplet, as described in section 2, and using consecutive triplets of views (in terms of movement), we can propagate the same type of motion from a triplet to the following one (as presented in figure 4) testing at each time the credibility of the synthesized view. The interested reader can find more details in [9] and examples of Mpeg encoded sequences of video camera simulated motions at *http://www.eurecom.fr/~image/spatialisation.html*

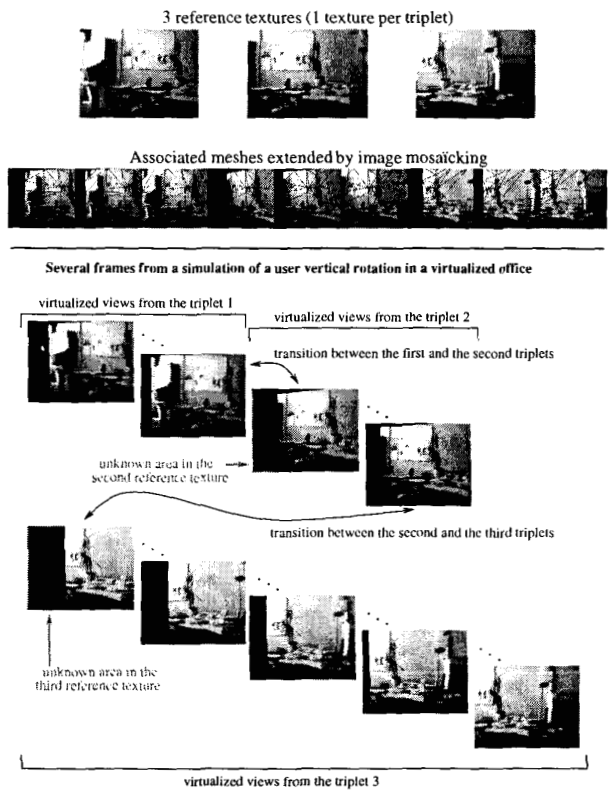


Figure 4. Synthesis of a user large rotation

These travelling simulations are fair but visually uncomfortable for the user, because of the transitions between the initial tri-view sequences. In fact, when switching from a triplet to the next or previous one, annoying visual artefacts are introduced in the virtualized picture. The last view generated from a triplet looks different from the first image synthesized from the following triplet: this is referred to as *triplets transition* (highlighted in figure 4). If we work on several triplets, image mosaïcking can be applied between

the synthesized output resulting image and the reference textures of the previous and the next neighboring triplets of images, in order to limit the non-informed area of the resulting image and to make up for the triplets transitions. The main idea here is to surimpose three images to render a realistic point of view, even if there is a switch of triplets:

- The usual intra-triplet synthesized view, really simulating $3D$ as described in section 2, is then displayed in front of two visual approximations, computed as explaned in the next step.
- The two approximations, denoted *first underlayer* and *second underlayer* in figure 5 are obtained by image mosaïcking between the reference texture of the previous and the current triplets for the first view, and the reference texture of the current and the following triplets for the second view. In practice, an underlayer is defined using a combination of two homographic transforms: an inter-triplet homography H_i between two reference textures (including the reference texture of the current triplet) just computed once and for all, and the intra-triplet homography H between the mesh of the reference texture of the triplet and the mesh corresponding to the current synthesized view (previous step), updated at each instant.

This method, entirely sketched in figure 5, is probably sub-optimal because the resulting views are composed by a virtualized image really simulating the user's $3D$ motion, on which attention should be focused, displayed over two approximated images called *underlayers* (more details can be found in [9]). These underlayers are obtained by operating image mosaïcking between the current virtualized view and the reference texture taken from the previous triplet of data for the first under-image and between the same virtualized view and the reference texture taken from the next triplet concerning the second underlayer. The underlayers are only approximations (except in the case of pure rotations) used to increase the visual comfort of the observer, making up for the non-informed areas of his virtualized point of view of the scene. Figure 6 presents visual results of the synthesis of virtualized points of view from altered trilinear parameters, with propagation on several triplets of pictures by image mosaïcking and underlayers method. With respect to figure 4 we note in particular that the mosaïcking steps to create the underlayers allow to increase the fluidity between consecutive triplets of views, smoothing the triplets transition between the last point of view generated from a triplet and the first one obtained from the following triplet. Such Mpeg encoded sequences are also available at *http://www.eurecom.fr/~image/spatialisation.html* for comparison with the previous sequences (i.e without mosaïcking procedure as shown in figure 4).

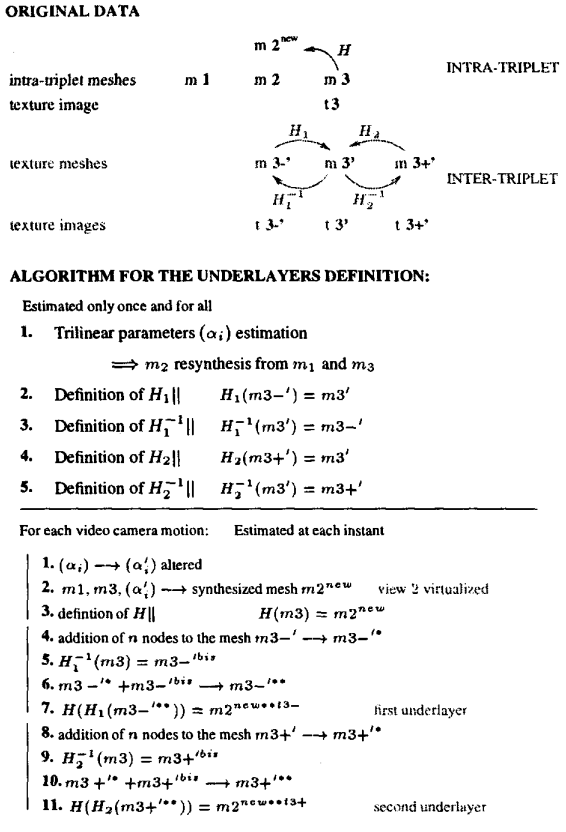


Figure 5. Triplets transition make up
m_i = intra-triplet meshes
m_i' = texture meshes, only defined from the view used as reference texture for a triplet (one texture mesh for a triplet) $\neq m_i$ for reasons of limited coverage
$m_i'-$ = previous texture mesh of m_i'
$m_i'+$ = following texture mesh of m_i'
$H \| H(m_i) = m_j$ = homographic transform from the mesh m_i to m_j
m_i^{new} = mesh m_i obtained by the simulation of a video camera motion
m_i^{*} = texture mesh m_i' extended by nodes addition

4. Conclusion

4.1. Future Work

In this paper, we have presented the extension on several triplets of views of our "mesh-oriented" approach for virtual views synthesis combined with an image-mosaïcking-based method, to increase the visual realism of *virtualized immersion*. This was called *video spatialization of a real* $3D$ *scene from multi-triplets of* $2D$ *views* introducing the concept of underlayers of a virtualized image, to offer the observer a better visual comfort.

Our future perspectives are focused on the integration of $3D$ objects in the $2D$ synthesized points of view of a scene. This is a necessary stage to offer users more interactivity in applications like virtual teleconferencing systems. But in this case, lots of problems like occlusions or collisions have to be studied (see figure 7).

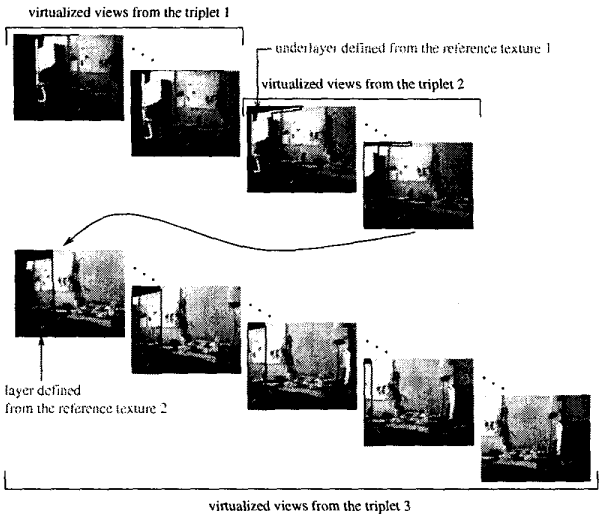


Figure 6. Minimization of the texture transitions using underlayers

The positions and orientations of the $3D$ objects inserted in the scene have to be coherent with the currently rendered $2D$ point of view of the user, unfortunately *a priori* unknown. Collisions between the inserted $3D$ objects and the initial objects present in the scene (only known by $2D$ images) have to be taken into account at each instant. To this extend, we have to define some main depth planes in the scene, to obtain a partial map of relative depth of the static objects of the scene. Our first investigations in this domain allow us to restore a discrete map of relative depths of the $3D$ scene, recovering perspective projection from the estimation of the trilinear parameters.

The complete management of $3D$ objects inserted in $2D$ spatialized environments is our future domain of interest, solving difficulties such as: differences of scale and lighting, and occlusions between objects of varying dimensionnality. Recent standards like MPEG-4 [13] consider this issue: one of the fundamental aims mentionned in the MPEG-4 SNHC call for proposals from the integration group is to "efficiently code interactive $2D$ and $3D$ environments consisting of real-time audio video and synthetic objects" [13]. The integration group experts focus on requirements for $2D/3D$ synthetic and natural data coding, seeking the integration of video coding (based on $2D$ feature analysis and model-based coding) and coding of structured $2D/3D$ graphical synthetic environments (including modeling, communication, run-time efficiency, real-time interaction and rendering of them), given the number of potential applications. Our image processing tool based on video spatialization is independent from the standard MPEG-4, but our work shares some of its major concerns and appears to be applicable in the context of an MPEG-4 encoder/decoder. The SNHC *mesh object* is a representation of a $2D$ deformable geometric shape, from which video objects may be created during a composition process at the decoder, by spatially piece–wise warping of existing video object planes or still texture objects. For this reason, video spatialization is an interesting technique to create a virtual environment without any explicit CAD model, using efficient image–rendering procedures for visualization.

Figure 7. Early insertion of $3D$ clones in a $2D$ environment

4.2. In the Context of a Televirtuality Project

Our work on video spatialization takes place originally in the larger TRAIVI[1] project, whose goal is to create a complete virtual teleconferencing system. In fact, the use of teleconferencing systems between multiple sites has considerably increased [14], because of industrial demands, but generally offers a poor quality of service [12]. The immersion of the participants in the same virtual and realistic environment, with the ability to move and look at the other participants, could make up for the lack of realism of classical systems and offer new ergonomic possibilities [11]. The TRAIVI project proposes an integrated approach that adresses both the participant's representation and the virtual meeting room background by mixing synthetic textured $3D$ face models with spatialized natural images. This virtualized vision of the real world is an alternative to the arbitrary artificial worlds, used in projects like [3] (where videos representing the participants of a videoconference are displayed in the virtual $3D$ model of a meeting area). The stake is then to render the real world in a way that is visually coherent and comfortable for its users.

Video spatialization for background control is one of the video processings we have to master in combination with

[1] TRAIVI stands for "TRAItement des images VIrtuelles" (Processing of Virtual Images)

model-based coding for participants control [20], to achieve a satisfactory level of visual realism in the development of a virtual teleconferencing system. That is why we focus on the synthesis of office or meeting-room images, with an emphasis for real-time visualization and realism of regenerated or unknown synthesized images, as opposed to the reconstruction accuracy. To that extent our "mesh-oriented" approach is a good trade-off in the context of the TRAIVI project, which requires realism and real-time.

The synthesis of virtual views is particularly interesting for the TRAIVI application: we can now imagine a virtual meeting composed of a pre-processing stage before the session. During this stage, information related to the user (his $3D$ model and his initial position) and the choice of the meeting area will be transmitted to a central site, which will pre-compute, from a few real uncalibrated views, the corresponding vectors of trilinear parameters and inter-triplets homography links, uploaded to each remote site. During the session, each site, independently from each others, will be able to create locally, by algebraïc processing applied on the trilinear parameters and intra-triplet homography, new coherent points of view for its user, based on his virtual position, motion parameters and center of interest in the meeting room, without sending any other information [5].

Our perspectives for the TRAIVI project are:

- the implementation of a complete room spatialization system, dealing with the quantity of pre-downloaded textures and the user's permitted motion granularity.
- the coherent integration of $3D$ models of the participants and background $2D$ images, which is still an open problem.

Acknowledgements The authors wish to thank Espri Concept for their support and contribution to this paper.

References

[1] D. Beymer, A. Shashua, and T. Poggio. Example Based Image Analysis and Synthesis. Technical Report 1431, MIT, 1993.

[2] J. Blanc and R. Mohr. Towards Fast and Realistic Image Synthesis from Real Views. In *SCIA'97*, Lappeenranta, Finland, 1997.

[3] C. Breitender, S. Gibbs, and C. Arapis. TELEPORT - An Augmented Reality Teleconferencing Environment. In *3rd Eurographics Workshop on Virtual Environments Coexistence & Collaboration*, Monte Carlo, Monaco, February 1996.

[4] J.-L. Dugelay and K. Fintzel. Image Reconstruction and Interpolation in Trinocular Vision. In *IMAGE'COM 96*, Bordeaux, France, Mai 1996.

[5] J.-L. Dugelay, K. Fintzel, and S. Valente. Synthetic Natural Hybrid Video Processings for Virtual Teleconferencing System. In *PCS*, Portland, Oregon, April 1999.

[6] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT PRESS, 1993.

[7] O. Faugeras and B. Mourrain. Algebraic and Geometric Properties of Point Correspondences between N Images. In *ICCV'95*, Boston,MA, June 1995. IEEE Computer Society.

[8] O. Faugeras and L. Robert. What Can Two Images Tell us about a Third One? *The International Journal of Computer Vision*, 1994.

[9] K. Fintzel and J.-L. Dugelay. a: Analytical Manipulations of Trilinear Parameters to Synthesize *a priori* Unknown Views, b: Initial Rotation Parameters Recovery from Trilinear Parameters of a Three Video Cameras System, c: Initial Perspective Projection Matrices Recovery from Trilinear Parameters of a Three Video Cameras System, d: Addition of Underlayers Visually Coherent to a Virtualized Image Obtained by Trilinear Synthesis. Technical report, EURECOM, Sophia Antipolis, France, 1997-98. in french.

[10] K. Fintzel and J.-L. Dugelay. Visual Spatialization of a Meeting Room from $2D$ Uncalibrated Views. In *IEEE IMDSP'98 Workshop*, Alpbach, Austria, July 1998.

[11] H. Fuchs, G. Bishop, K. Arthur, L. McMillan, R. Bajcsy, S. Lee, H. Farid, and T. Kanade. Virtual Space Teleconferencing using a Sea of Cameras. In *First International Symposium on Medical Robotics and Computer-Assisted Surgery 2*, pages 161–167, Pittsburgh, Pa, September 1994.

[12] K. Jeffay, D.-L. Stone, T. Talley, and F.-D. Smith. Adaptive, Best effort Delivery of Audio and Video Across Packet-Switched Networks. In 3^{rnd} *Intl. Workshop on Network and OS Support for Digital Audio and Video*, SanDiego, CA, November 1992.

[13] MPEG–4 Synthetic/Natural Hybrid Coding. URL http://www.es.com/mpeg4-snhc/

[14] K. Okada, F. Maeda, Y. Ichikawaa, and Y. Matsushita. Multiparty Videoconferencing at Virtual Social Distance: Majic Design. In *ACM'94*, pages 385–393, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223 Japan, October 1994.

[15] B. Rousso, S. Avidan, A. Shashua, and S. Peleg. Robust Recovery of Camera Rotation from Three Frames. In *CVPR'96*, June 1996.

[16] A. Shashua. On Geometric and Algebraïc Aspect of $3D$ Affine and Projective Structures from Perspective $2D$ Views. In J.-L. Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of Invariance in Computer Vision*. Second European Workshop Invariants, Ponta Delagada, Azores, October 1993.

[17] A. Shashua. Projective Structure from Uncalibrated Images: Structure from Motion and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):778–790, August 1994.

[18] A. Shashua. Trilinearity in Visual Recognition by Alignment. In *ECCV A*, pages 479–484, 1994.

[19] M.-E. Spetsakis and J. Aloimonos. A Unified Theory of Structure from Motion. In *DARPA IU Workshop*, pages 271–283, 1990.

[20] S. Valente and J.-L. Dugelay. Face Tracking and Realistic Animations for Telecommunicant Clones. In *ICMCS'99*, Firenze, Italy, June 1999.

[21] D. Watson. *A Guide to the Analysis and Display of Spatial Data*. Pergamon PRESS, 1992.