Appears in: *ECMAST, 21 - 23 May 1997, Milan, Italy*

A Multi–Site Teleconferencing System using V.R. Paradigms

Stéphane Valente & Jean–Luc Dugelay

Multimedia Communications Department
Institut EURÉCOM
2229, route des Crêtes
B.P. 193
F-06904 Sophia–Antipolis Cedex

Tel.: +33–(0)4.93.00.26.66
Fax: +33–(0)4.93.00.26.27
E–mail: {valente,dugelay}@eurecom.fr
URL http://www.eurecom.fr/~image

Abstract

This paper discusses early results in a televirtuality project named "TRAIVI". The goal of this project is to create and run multi–site meetings via low bit–rate links and virtual reality paradigms. We propose to enable several persons located at different physical sites to meet each other in a common virtual meeting room. In order to preserve a high level of realism, we describe how it is possible to animate $3D$ human–like synthetic interfaces based on "CYBERWARE" models, and how they can be immersed in a common virtual meeting space created from a limited set of real room pictures.

1 Introduction

Low bit–rate video communication systems can be divided into two categories — as they can be signal–processing oriented (meaning that they consider video images as $2D$ or $3D$ stochastic signals, and try to efficiently use the signal redundancy to obtain a low–bit rate coding of the images) — or object–oriented (they consider an image as a $2D$ projection of $3D$ physical objects, and try to encode the scene objects and how they are displayed on an image). Both approaches differ in their results: the first one aims at coding and decoding the real image with as much visual fidelity as possible, usually measured by a MSE criterion, whereas the second one aims at rebuilding more

or less accurately the image so that it looks coherent and realistic. We explain in this introduction that thanks to *Virtual Imaging* techniques, in the case of multiple site teleconferencing, model–based coding can outperform the signal–oriented approach.

1.1 Current Signal–Oriented Systems Limitations

Presently, multi–speaker teleconferencing systems are satisfactory for meetings involving no more than two people at the same time, or more generally, no more than two sites. Beyond this point, human communication can become critical, compared to real meeting conditions, because you have to alternatively display the view of the currently speaking site (due to bandwidth limitations), or if you have more bandwidth, you have mini–images representing each site. It is clear that if you see the meeting as multiple views like figure 1(a), you cannot feel that you are *with* the other participants. And if you have only the view of the talking site, users must be self-disciplined and speak when they are "on display" if they want the meeting to keep a certain level of intelligibility. Furthermore, such systems suffer from important limitations, due to the difficulty to achieve convincing eye contact between speakers, and the persistent feeling of distance because of the lack of common meeting room references and poor audio immersion.

(a) Classical View. (b) Virtual View.

Figure 1: What a fourth participant would see during a multisite teleconferencing session.

1.2 Virtual (Model–Based) Teleconferencing

The concept of virtual teleconferencing offers elegant solutions to the problems of classical teleconferencing systems. The key idea is to provide speakers with a common meeting space as if they were all meeting in the same physical room, to synthesize the individual points of view they would naturally experience, and to give them the opportunity of having eye contact with each other via synthetic $3D$ models of the participants (clones), in short, to obtain a synthetic view like figure 1(b). In order to achieve a good level of realism, several audio and video techniques have to be investigated and implemented. Among them are audio spatialization and multiplexing, echo cancelation, video spatialization, speakers face cloning, audio–video synchronization ... in order to get a perfect immersion into the meeting space.

1.3 Related Work

A virtual teleconference system is in fact related to two video processing issues found in the literature (aside networking aspects): being able to reproduce the participants images, and providing a virtual environment to immerse those images with the real users.

In recent years, research has been conducted on topics related to teleconferencing and model–based coding such as human image analysis and reproduction [TW93], and human face cloning [SVG95], that can be useful to reproduce the participants. Historically, all facial animation research works on the basis of a generic face model (the well known "CANDIDE" model is the most common), activated by a few control nodes according to the FACS system [EF77]. Whereas such an approach might give unrealistic faces as results [EDP94], more and more research teams try to improve the synthesized generic models: the INA (Institut National de l'Audiovisuel, France) builds a special texture to be mapped onto the face model [Ins], Reinders *et al.* adapt a generic face model for individuals [RvBSvdL95], and Choi *et al.* obtain amazingly real expressions with their textured model [CAHT94]; however, the inverse idea has not been investigated yet in the literature: start with a model depending on a person, and make it more generic so that it can be handled by an automatic framework. This is what we propose to do in our research work on face cloning.

As far as building virtual environments is concerned, some experiments have also been done via the TELEPORT project with a wall–sized display. It uses a synthetic $3D$ scene that carefully matches the room in which the display is located, and video images of remote participants are blended into the virtual extension of the room [GDML]. Once again, instead of building a special $3D$ environment, we would like to start from existing ones to be closer to what users are used to in the real world.

1.4 The TRAIVI Project

The TRAIVI[1] project aims at implementing virtual meeting rooms over low–rate bindings with a high level of realism. Some authors, like Kanade [KNR95], made clear that *virtualized reality* is superior to *virtual reality* since it takes into account the real world fine details and not a simplistic CAD model. This is the reason why we want to use $3D$ textured wire frame models scanned from real people ("CYBERWARE" models) which are everything but generic to perform the face cloning. Our primary goal is neither to build an imaginary world that has no equivalent in reality, nor to exactly synthesize the real world and the true speakers expressions, but to render the real world in a way that is visually coherent and *comfortable* for its users. The same philosophy stands true for the meeting space environment with video spatialization: this technique uses real room uncalibrated pictures and *virtualizes* them to create the meeting room views, as opposed to building $3D$ room models from scratch.

Some European projects, namely HUMANOID [BCH+95] and its continuation VIDAS [VID], are investigating interpersonal audio/video communication using virtual reality paradigms, their two main objects being also the $3D$ talking interfaces and the

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

background, each of them optionally natural, synthetic or hybrid. Our approach can be compared to their "natural" methodology.

This paper presents how our clone models are handled both globally (their pose in the $3D$ space) in section 2, locally (their facial expressions) in section 3, and how they can be placed in a virtual environment designed by video spatialization in section 4.

2 Global Animation

Global animation consists in determining the speaker's pose in the $3D$ space by image analysis techniques in real–time, and to use the extracted parameters to render the clone in a coherent manner as in figure 2. Our analysis scheme achieves it without requiring to tape marks on user's faces, and without user–assistance during the initialization stage.

In this section, we will briefly describe what a CYBERWARE model is, which parameters are analyzed from the video input, and how they are tracked.

2.1 CYBERWARE Models

CYBERWARE models are produced by three–dimensional cylindrical scanners, and arrive in 2 files: the first one contains a set of $3D$ coordinates representing the scanned head geometry (i.e. a wire frame), and the second one holds texture information to be mapped to the geometrical coordinates.

These models are highly realistic, and contain precise information about the participant who was scanned. This can turn out to be a drawback, because each model is valid only for one person and a static facial expression, and lacks generality. Besides, the mesh model is unoptimized in terms of $3D$ nodes number, roughly 1.4 million. The face $3D$ surface sampling was made along regular cylindrical coordinates, and as a result, there are as many points to represent smooth surfaces (like the cheeks) as to represent the sharpest ones (like the nose). It has neither anatomical knowledge (like bones and muscles under the skin to model human face deformation possibilities) nor a physical model (to be able to deform it along the time axis) [TW93]. In order to be capable of local deformations, further processings are under investigation to alter the plain $3D$ node set (see section 3.2).

2.2 Extracted Parameters

To derive the degrees of freedom of the head global motions, we use the parameters obtained by image analysis and feature tracking showed in figure 2: the face outline is materialized by the rectangular window W, and within the window, we detect the eyes L and R, the eyes horizontal axis H and the vertical one V. The head degrees of freedom are then evaluated by:

left/right and up/down translations:	given by the window W center coordinates
forward/backward translation:	derived from the width of W
left/right rotation:	given by the position of V within W
up/down rotation:	given by the position of H within W
last rotation degree:	given by the angle between the eye positions L and R

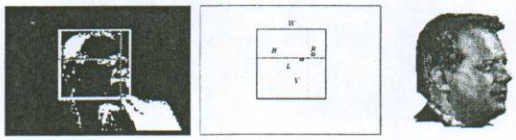


Figure 2: Global motion analysis parameters.

2.3 Analysis and Tracking Algorithm

We are now going to explain how the parameters in figure 2 are estimated in a video sequence. We proceed in two steps: first we determine the head outline (refered as window W in the figure) and then, all other parameters by tracking the speaker's eyes within W.

2.3.1 Head Outline Determination

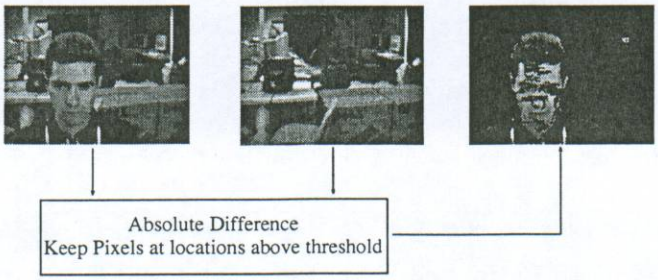


Figure 3: Speaker's outline thresholding.

Our head outline tracking algorithm assumes that the background behind the speaker remains static throughout the session. This enables us to simply subtract the current speaker image from a background reference view and threshold the difference to get the outline (figure 3); then, we compute the binary horizontal and vertical histograms to find the top, left and right edges of the head (figure 4). We do not look for the bottom edge of the head, since it is not necessary regarding the parameters mentionned in section 2.2. The static background assumption is acceptable for this application because it leads to a simple and real–time algorithm. Besides, there is no pint to preserve the speaker's background during the global motion analysis process because his clone will be rendered in a different environment.

This algorithm has been extended to achieve tracking by using the formerly found values for a new video frame: the previous window W is slightly widened and used as a search region to compute the binary histograms. If the head edges cannot be found, the search region is extended again, and ultimately, the whole frame is searched.

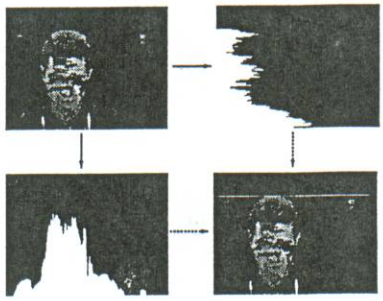


Figure 4: Speaker's outline binary histograms.

2.3.2 Eyes Tracking

The H, V, L and R parameters are derived from the position of the eyes. The eyes tracking algorithm must face three distinct challenges: it must be robust to scale changes (when the user moves forward or backward in front of the camera); to pattern modifications (when the speaker moves his head, the projected eye pattern is altered); and finally, it must be robust to the face illumination changes (when the speaker moves relatively to the scene light sources). The last two challenges can be summed up by saying that the algorithm must cope with eye pattern geometric or photometric changes. This section describes a two pass template matching algorithm that meets these requirements.

Basic Template–Matching Formulation

Template–Matching is widely used as a pattern recognition tool to determine a similarity measure between a reference pattern and an unknown pattern. A template image representing the reference pattern is applied onto a bigger image. At each point, a match score is computed to create an intermediate score image, and its peaks indicate the positions where there is a high match between the unknown pattern and the reference one.

Although most template–matching algorithms consider template images with their origin being centered, we propose to use a slightly different formulation to allow the template origin not to be centered. From now on, we will refer to the template origin as the *template hot spot*. This can become very useful to include more discriminative patterns without over–increasing the template size: let us imagine that we want to detect the left corner of the left eye. The template should contain a reasonable amount of highly–contrasted details, for instance the iris and the eyebrow. The hot spot is here in the lower–left quarter of the template. If the hot spot was constrained to be the template image center, the template image would obviously have to be bigger to contain the same details.

Let $I(i,j)$ denote the tested image of size $N \times M$ with $0 \leq i < N$ and $0 \leq j < M$, and $T(i,j)$ the template of size $O \times P$ with $0 \leq i < O$ and $0 \leq j < P$. Then, given a cost function $f(x,y)$ and the template hot spot position (h_x, h_y), the score $S_f(k,l)$ at

the image point $I(k,l)$ is

$$S_f(k,l) = \frac{1}{O \times P} \sum_{i=0}^{O-1} \sum_{j=0}^{P-1} f(I(k+i-h_x, l+j-h_y), T(i,j))$$

The traditional cost functions are the *absolute differences* $f_{AD}(x,y) = |x-y|$, the *squared differences* $f_{SD}(x,y) = (x-y)^2$ and their mean intensity level invariant forms, the *normalized absolute difference* $f_{NAD}(x,y) = |x-y-\mu_d|$ and the *normalized square difference* $f_{NAD}(x,y) = (x-y-\mu_d)^2$ where

$$\mu_d = \frac{1}{O \times P} \sum_{i=0}^{O-1} \sum_{j=0}^{P-1} I(k+i-h_x, l+j-h_y) - T(i,j)$$

is the mean of the image/template pixelwise difference.

Adaptation to Varying Patterns

We addressed the problem of eye pattern modification by introducing dynamic templates: for each eye position that is found, we run confidence tests to assess the likelihood of the position based on the similarity score between the eye and its template and the position of the eyes within the window W (see figure 2). If the current position satisfies these tests, the template patterns are updated with the current eye images, and therefore adapt automatically to any change of eye shapes or illuminations. Parallely, the window W is used as an indication of the face scale to modify the template size. This way, the templates tend to keep the same amount of significant details and do not catch parasite details: if the initial templates contained the eyes and eyebrows, and the user moves away from the camera, the templates will be updated with smaller image portions still displaying the eye and eyebrows, and will not include the user's nose or ears which could later attract the templates.

The similarity measure cost function is also dynamically adapted, depending on the level of certainty we have about the current eye pattern: during the initialization stage, when the eyes are searched for the first time, there is little chance that the first eye templates have the same mean intensity as the current speaker image because of the scene unknown lighting conditions. The first similarity measure we use is the sum of normalized absolute differences S_{NAD}. During the session, if the templates have just been updated, we can assume that the mean intensity of the eyes did not change between two consecutive frames, and we use the sum of the absolute differences S_{AD}. This measure is quicker to compute, and it helps to track the eyes not only by considering the eye patterns, but also by considering the mean intensity of the eyes. As soon as the eyes tracking process leads to bad similarity scores, indicating that the mean intensity level may have changed, or as the templates were not updated because of inconsistent temporal redundancy of the best score positions, we revert to the mean intensity level invariant measure S_{NAD}.

Recentring the Dynamic Templates

Dynamic templates are likely to deviate from the feature they are supposed to track, due to the fact that they are constantly updated: the new template pattern is extracted from the current frame relatively to the best match position. If the best match does not occur on the exact eye center, the eyes will gradually disappear from the dynamic templates (figure 5). To make the updated templates more reliable, it is necessary to precisely locate the eyes centers through a second pass of template–matching. This recentring procedure is run with a reference eye center pattern which is simply scaled (as opposed to updated) to fit the current face scale.

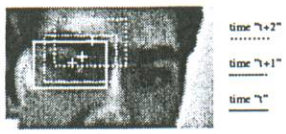


Figure 5: Dynamic templates deviation.

Figure 6 provides a few examples of eyes tracking, regardless to the speaker's pose. The first dynamic templates contained the user's eyes and eyebrows, and the recentring templates only the irises: the reader can notice that thanks to the eyebrows, the eyes can still be found even if they are closed.

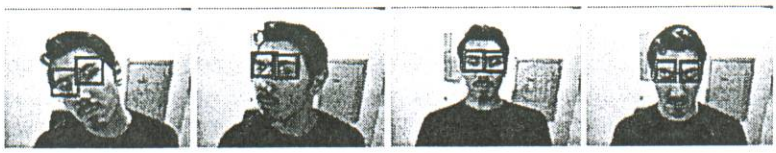


Figure 6: Dynamic template matching robustness to $2D$ rotation, $3D$ rotation, scale changes and closed eyes.

2.4 Global Animation Demonstration Software

Our analysis software validates the algorithms described in this section, and demonstrates the eyes tracking robustness. The head outline determination still needs to get improved, as it can fail to track the head when the global scene illumination changes, most often due to the variable sunlight coming through the office windows. Nevertheless, the static assumption background proved to be relevant in most cases, and it can easily be enforced by an algorithm which updates the reference background image whenever it is necessary. The software can sustain a rate of roughly 10 frames per second on an SGI Indigo Indy workstation with 208×160 luminance images (or 7 frames per second with 320×242 images).

On the synthesis side, the "CYBERWARE" mesh was processed to obtain 3000 triangles making the model become practical for real–time display on a SGI High Impact

workstation (see section 3.2 and figure 8). The rendered head is controlled by six degrees of freedom, and rendered in front of a background image, which will be later on replaced by virtual views produced by video spatialization (see section 4.2).

The audio connection between the two workstations is realized by separate processes via UNIX sockets. Later on, it will be incorporated within the analysis and synthesis software, when audio/local animation synchronism becomes necessary.

The next step for global animations will be to implement a kind of $3D$ motion model coupled with a Kalman filter to make the global motion recovery even more robust.

3 Local Animation

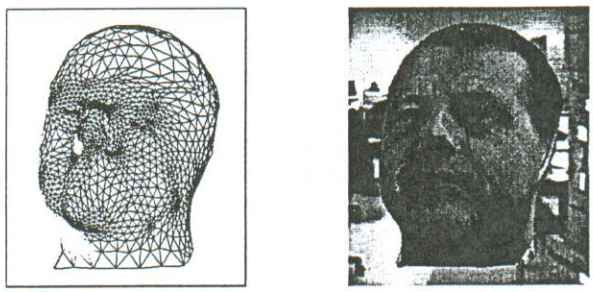
Global animation just controls the position of the model in the $3D$ space without changing its facial expression. A local animation scheme has to be implemented in order to provide the user with an interface displaying the other participants facial expressions. As it has been mentionned previously in section 2.1, a plain "CYBERWARE" model has some disadvantages, in terms of nodes number and little anatomical and physical knowledge, unlike "hand–made" models. Several teams work on the basis of a "CYBERWARE" model, but they all adapt it to a physics–based model: Terzopoulos and Waters adapt a generic mesh model composed of several physics–based tissue layers and muscle actuators to the "CYBERWARE" data [TW93], and Essa, Sclaroff and Pentland propose to fit a superquadric object with displacement maps to the range data to add elastic properties via the finite element theory [ESP93].

Our primary goal to perform local animation is not to implement a physics–based model onto the "CYBERWARE" data, but to rely on efficient computer graphics techniques to cope with real time constraints. Two different strategies are currently being investigated: the first one consists in simulating animation by altering the texture attached to the wire frame (section 3.1), and the second one in manipulating the wire frame itself (section 3.2). It is important to realize that different facial features might require different animation procedures. Our current work focuses on the local synthesis aspects of the model rendition because it will decide for each facial feature which analysis techniques and parameters are the most suitable to tie the animation to a video sequence via a low bandwidth communication line (section 3.4).

3.1 Textural Animation

We tried to animate the eyes first because we already know how to track them. The point of textural animation is to map different textures onto the wire frame at rendition time. We have already implemented routines in our synthesis software that dynamically switch between several eye textures (figure 7) representing three distinct gaze directions, with almost no computation overload in our synthesis software. The middle texture is the original data produced by the "CYBERWARE" scanner, and the two others were pre–calculated using commercial image editing products.

Figure 7: Gaze control via texture modification.



(a) (b)

Figure 8: Mesh and textured model.

3.2 Wire Frame Animation

Other features, like the eyebrows or wrinkles, can be animated either by texture modification, or by wire frame control.

We transformed the wire frame using Delingette's approach [Del94]: it is adaptively remodelled to adapt the mesh complexity to the surface geometry, to get approximatively 1,400 nodes as displayed on figure 8(a) (starting from more than 1 million). This kind of active mesh supports rendering via simplex interpolations, which can lead to accurate expression wrinkles [Via92]. Further work has to be done to relate the processed wire frame to *Facial Action Coding System* (FACS) real–time software control units.

3.3 Combination

Textural animation is both a low–cost and powerful alternative to wire frame animation, and a palliative technique to active mesh animation limitations: let us consider for instance the problem of animating the $3D$ model jaws. If the user opens his mouth, his teeth and tongue will become visible, but unfortunately, these features are not part of the "CYBERWARE" data (because the scanner digitalizes only the face surface, and not its inner parts). It will be necessary to resort to portions of real images to make them appear on the synthesis side to preserve the level of realism.

We then have a choice between using live portions of images, or pre–defined textures. We think that resorting to live images should be avoided as much as possible for three reasons: firstly, lively extracted images are likely to be valid only under the camera

point of view, whereas the synthesis side should still be able to render the clone under any point of view; this is not applicable before we find the right $2D \rightarrow 2D_{cylindrical}$ transformation; secondly, lively extracted images cannot be put directly into the "CYBERWARE" texture because they do not have the same photometric and geometric properties and lightning conditions; and finally, the bandwidth requirements could drastically inflate due to the transmission of images portions.

Building pre–defined texture dictionaries (possibly from typical teleconferencing sessions images) and referencing indexes currently seems to be the most efficient solution.

3.4 Prospective Studies

Once the synthesis policy is set in our demonstration software for each animated feature, we will need local analysis techniques to feed the animation according to the video input.

As far as textural animation is concerned, *Image Matching* seems to be the most appropriate technique to find the best pre–defined texture for the eyes (see section 3.1), or the user's forehead wrinkles, if they are simulated by "texture only". If necessary, the similarity measure can also be made scale and rotation invariant [CJ93].

Manipulating the wire frame is a far more complex task. Although our "CYBERWARE" model cannot be used "as–is", it offers a nice property compared to other models: its precise mapping between texture and $3D$ geometry, allowing a high level of realism. We would like to take advantage of this realism with an analysis/synthesis cooperation. There is a very interesting possibility to create it off–line with *eigenfeatures*. Usually, eigenfeatures are used for their probabilistic learning capability: they give a compact representation of a rather complex space (like the space spanned by a training set of face images) by finding the set of orthogonal vectors that represent the maximum energy subspaces. They have been widely used to recognize faces as a discriminative measure [PMS94], and an attempt to make them classify poses is found in [DMP96].

The main difficulty to compute eigenfeatures is to set up a good training database. The features have to be well–scaled, well–centered, and the lightning conditions constant, which is far from being easy if out–of–the–lab images are exploited. This is where the "CYBERWARE" model can take its full modelling power: it can accurately be rendered under any pose, and with any facial expression defined by the synthesis software, and with known control parameters that produced the given facial expression. In this case, not only will we get optimally trained eigenfeatures, but also they will come in an optimal basis to map the facial expressions found in images to the right control parameters to activate the wire frame. This is the research direction we are to investigate first.

And finally, in the case the speaker's face is not turned towards the camera, it will also be quite acceptable to apply a realistic (but not real) deformation on the clone lips based on the audio signal analysis [Ben96].

4 Virtual Environment Aspects of the Project

The second aspect of the TRAIVI project focuses on the virtual environments creation problem, and the scenarios that will control the clone insertion.

4.1 Speaker's Virtual Positions

Each participant should have the possibility to choose his own position within the meeting according to his center of interest. Let us assume that for instance, user 1 wants to *virtually* sit in front of user 3, and have users 2 and 4 stand respectively on his left and right hand side, leading to the disposition of figure 9. The stake is now to synthesize a scene that recreates the point of view that user 1 would have if the meeting actually took place with this disposition, like in figure 1(b).

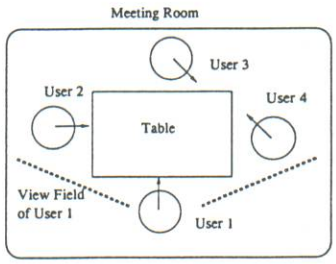


Figure 9: Virtual speakers positions chosen for site 1 (see also figure 1(b)).

Three issues are then related to this purpose:

1. To improve the feeling of being present in the same physical room, the clones will be displayed on personalized backgrounds on wall–sized displays. The system has then to render background images that will be coherent with the speakers' position, point of view and field of view they would experience in the real world given the meeting disposition. This question is addressed by *Video Spatialization* techniques (introduced in section 4.2). Please also note that this approach allows us to get rid of the real background image during the outline determination step exposed in section 2.3.1.
2. Advanced audio techniques have to be investigated to provide users with spatial cues coherent with the meeting disposition (section 4.3). These post–processings will be implemented at destination sites, in order to match the virtual local speakers' disposition.
3. Networking components must be defined to transparently exchange the meeting common data and the models (section 4.4).

4.2 Video Spatialization for Virtualized Meeting Room

Although rendering a meeting room under any point of view in real–time for the clones background is quite feasible with a CAD model, realism considerations lead us to study

video spatialization which proposes to virtualize a set of real room images to quickly access any point of view.

In this framework, no knowledge about the pick–up equipment is assumed: neither internal nor external calibration parameters, so that it allows the system to use uncalibrated views taken by low–cost video cameras. An effective algorithm based on trilinearity constraints able to reconstruct an existing view from two other neighboring views and to create new *virtualized* points of view has already been proposed for our project in [DF96]. It was developed in the limited context of three different views, and it has to be extended to more reference views to be able to reconstruct an entire meeting space. Interested readers are kindly invited to report to this publication for deeper details. Figure 10 shows a few examples of virtualized views.

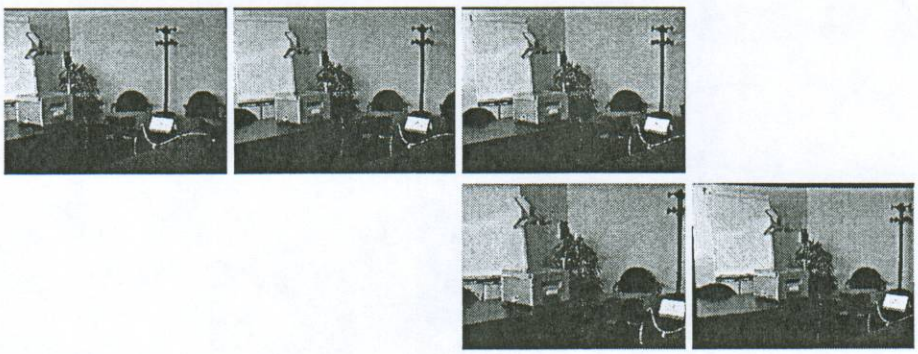


Figure 10: From left to right, top to bottom: left, center and right reference existing views — and zoomed and shifted right synthesized views by *Video Spatialization*.

4.3 Audio Immersion

In many (if not all) classical teleconferencing systems, audio aspects are often neglected because efforts are mainly focused on the video compression algorithms, and people generally assume that the visual quality is much more important that the audio quality. However, audio spatialization techniques can improve the immersion feeling by giving the users more spatial cues, and also improve the general meeting intelligibility thanks to the *Cocktail Party Effect*, allowing several people to talk at the same time [Jot96].

4.4 Networking Aspects

The system that will be implemented looks like figure 11. Since the users must be provided with a realistic and coherent view of the meeting situation, the sites will transmit only parameters that will remain meaningful under any participant's point of view.

In the current implementation involving two sites and processing only the 6 degrees of freedom, the required bandwidth can be estimated at a little bit more than 64 kbits per second in each direction (regardless of IP protocol encapsulation overload):

Video Parameters: 6 parameters × 2 bytes/parameter × 10 frames/second
= 120 bytes/second
Audio Parameters: 8000 samples/second × 1 byte/sample (μ–law encoding)
= 8 kbytes/sec

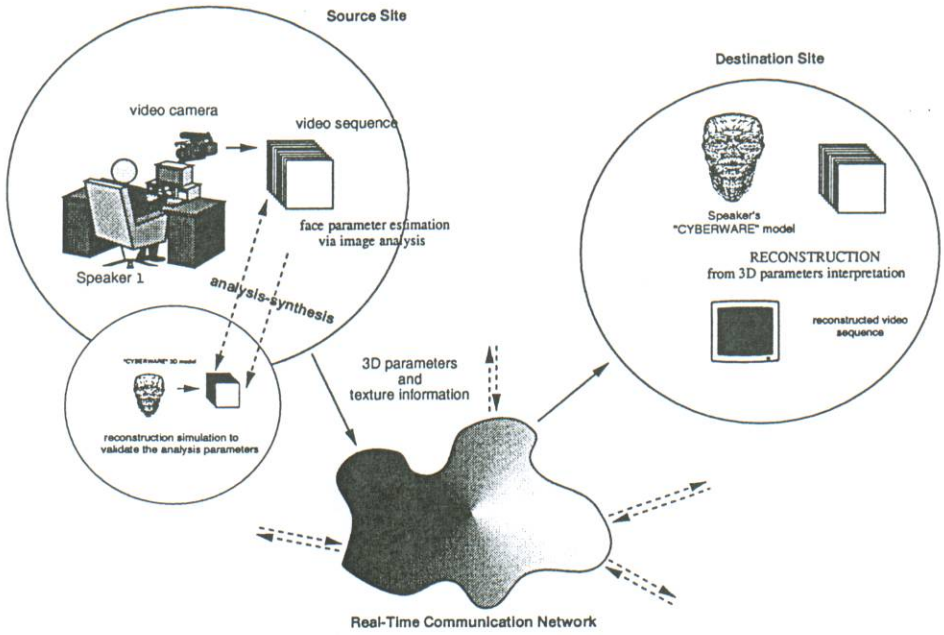


Figure 11: Virtual teleconferencing system configuration

$3D$ parameters are extracted at the source site, sent to destination site(s), and finally interpreted to render the $3D$ scene under another point of view. Audio post–processings are also to take place at destination sites later on.

The teleconferencing system is partially illustrated on figure 11: more than two sites can participate in the meeting. Furthermore, a central site in the network will have the responsibility to transmit to all teleconferencing sites the common meeting room data, the participants virtual locations around the table, and the right "CYBERWARE" models (if they are not found in local databases) before the meeting begins. The protocol and network components that would exchange data between all sites have still to be defined.

4.5 Future Possibilities

We have presented a meeting scenario that conforms to real meeting conditions, enabling the participants to share the same representation, but one can imagine stepping beyond real meeting constraints, and fullfilling more of user requirements:

- users could set up their own local meeting disposition, independently from each other. Every participant would have the possibility to sit in front of the meeting

moderator, and the system would animate the clones in such a way that people still know who is looking at who and are allowed to eye contact the participants. The left/right rotations would no longer correspond to the real participants global motions in order to direct the clones towards the person they are virtually looking at.

- instead of sharing the same virtual meeting room environment across all participants, users could take a few shots of their own office room, virtualize it, and insert the clones in it as if the meeting was taking place locally in their office.

5 Summary and Conclusions

We have discussed the TRAIVI project and the associated video cloning and video spatialization issues, focusing more on virtual*ized* reality than virtual reality.

Video Cloning aims at providing people with $3D$ interfaces within a virtualized meeting environment. "CYBERWARE" models are used to ensure a high level of realism. We discussed the specificities of such models, and how they can be efficiently animated, regardless of the point of view under which they are rendered. Whereas the global animation scheme is mature, the local animation procedure is currently under development though some partial animation has been achieved.

We have also presented the virtual environment creation strategy, integrating audio and video features. *Video Spatialization* is hoped to help reach new standards in virtual worlds, that have been produced only by CAD methods until now. Through this technique, some virtual points of view, such as virtual focal length changes, can be accessed. Some further investigations are in progress to finely control all physical transformations (video camera virtual translations and/or rotations), the ultimate goal being to be able to obtain a set of overlapping views that would cover the entire meeting space where the clones are to be inserted.

References

[BCH+95] R. Boulic, T. Capin, Z. Huang, L. Moccozet, T. Molet, P. Kalra, B. Lintermann, N. Magnenat Thalmann, I. S. Pandzic, K. Saar, A. Schmitt, J. Shen, and D. Thalmann. The HUMANOID environment for interactive animation of multiple deformable human characters. In *Eurographics'95*, Maastricht, Netherlands, 1995.

[Ben96] C. Benoit. On the production and perception of audio-visual speech by man and machines. In Y. Wang, S. Panwar, S.-P. Kim, and H. L. Bertoni, editors, *Multimedia Communications and Video Coding*, pages 277–284. Plenum Press, New–York, 1996.

[CAHT94] C. S. Choi, K. Aizawa, H. Harashima, and T. Takebe. Analysis and synthesis of facial image sequences in model–based image coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(3):257–275, June 1994.

[CJ93] G. S. Cox and G. de Jager. Template matching with invariance. In *Proceedings of the Fourth South African Workshop on Pattern Recognition*, pages 152–156, 1993. URL http://dip1.ee.uct.ac.za/papers/cox93.html.

[Del94] H. Delingette. *Modélisation, Déformation et Reconnaissance d'Objets Tridimensionnelles à l'aide de Maillages Simplexes*. PhD thesis, Ecole Centrale de Paris, Châtenay–Malabry, France, 1994.

[DF96] J.-L. Dugelay and K. Fintzel. Image reconstruction and interpolation in trinocular vision. In *IMAGE'COM*, pages 277–282, Bordeaux, France, May 1996.

[DMP96] T. Darell, B. Moghaddam, and A. Pentland. Active face tracking and pose estimation in an interactive room. Technical Report 356, M.I.T. Media Laboratory Perceptual Computing Group, 1996.

[EDP94] I. A. Essa, T. Darrell, and A. Pentland. Tracking facial motion. In *IEEE Workshop on Nonrigid and Articulate Motion*, Austin, Texas, November 1994.

[EF77] P. Ekman and W. V. Friesen. *Facial Action Coding System*. Consulting Psychologists Press, Palo Alto, California, 1977.

[ESP93] I. A. Essa, S. Sclaroff, and A. Pentland. Physically–based modeling for graphics and vision. In Ralph Martin, editor, *Directions in Geometric Computing*. Information Geometers, UK, 1993.

[GDML] GMD's Digital Media Lab. TelePort: The Communication Wall. URL http://viswiz.gmd.de/DML/cwall/cwall.html.

[Ins] Institut National de l'Audiovisuel. Televirtuality project: Cloning and real-time animation system. URL http://www.ina.fr/INA/Recherche/TV.

[Jot96] J.-M. Jot. Synthesizing three–dimensional sound scenes in audio or multimedia production and interactive human–computer interfaces. In *L'Interface des Mondes Réels & Virtuels*, Montpellier, France, Mai 1996.

[KNR95] T. Kanade, P. J. Narayanan, and P. W. Rander. Virtual*ized* reality: Concepts and early results. In *IEEE Workshop on Representation of Visual Scenes*, Cambridge, Massachusetts, June 1995. In conjonction with ICCV'95.

[PMS94] A. Pentland, B. Moghaddam, and T. Starner. View–based and modular eigenspaces for face recognition. In *International Conference on Computer Vision and Pattern Recognition*, June 1994.

[RvBSvdL95] M.J.T. Reinders, P.L.J. van Beek, B. Sankur, and J.C.A. van der Lubbe. Facial feature localization and adaptation of a generic face model for model–based coding. *Signal Processing: Image Communication*, 7:57–74, 1995.

[SVG95] A. Saulnier, M.-L. Viaud, and D. Geldreich. Real–time facial analysis and synthesis chain. In *International Workshop on Automatic Face— and Gesture—Recognition*, pages 86–91, Zurich, Switzerland, 1995.

[TW93] D. Terzopoulos and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6), June 1993.

[Via92] M.-L. Viaud. *Animation Faciale avec Rides d'Expression, Vieillissement et Parole*. PhD thesis, Université de Paris XI–Orsay, Orsay, France, 1992.

[VID] VIDAS — Video assisted audio coding and representation. URL http://www.uni-stuttgart.de/SONAH/Acts/AC057.html.