The Visual Computer

Hyper mask – projecting a talking head onto a real object

T. Yotsukura[1], S. Morishima[1], F. Nielsen[2], K. Binsted[3], C. Pinhanez[4]

[1] Faculty of Engineering, Seikei University, 3-3-1 Kichijoji-Kitamachi, Musashino-shi, Toyko 180-8633, Japan
E-mail: {yotsu,shigeo}@ee.seikei.ac.jp
[2] Sony Computer Science Laboratories Inc., 3-14-13 Higashi-Gotanda, Shinagawa-ku, Tokyo 141-0022, Japan
E-mail: nielsen@csl.sony.co.jp
[3] I-chara Inc., 2-34-1 Uehara, Shibuya-ku, Tokyo 151-0064, Japan
E-mail: kimb@i-chara.com
[4] IBM T.J. Watson Research, Route 134, P.O. Box 218, Yorktown Heights, N.Y. 10598, USA
E-mail: pinhanez@us.ibm.com

Published online: ?? ?? 2002

HyperMask is a system which projects an animated face onto a physical mask worn by an actor. As the mask moves within a prescribed area, its position and orientation are detected by a camera and the projected image changes with respect to the viewpoint of the audience. The lips of the projected face are automatically synthesized in real time with the voice of the actor, who also controls the facial expressions. As a theatrical tool, HyperMask enables a new style of storytelling. As a prototype system, we put a self-contained HyperMask system in a trolley (disguised as a linen cart), so that it projects onto the mask worn by the actor pushing the trolley.

Key words: Talking heads – Homography – Neural networks – Computerized theatrical performances – Lip-synch

Correspondence to: T. Yotsukura

The Visual Computer (2002) 18:???–???
Digital Object Identifier (DOI) 10.1007/s003710100140

1 Introduction

HyperMask is a demonstration technology for a theatrical tool. It enables a new style of storytelling, in which a human actor's performance is enhanced by the system in an entertaining manner. However, the same technology could also be useful for other applications in which active projection is necessary. For example, in the so-called "Office of the Future" (Raskar et al. 1998) or an interactive playground, we would like to be able to project dynamically images and information onto moving, irregularly shaped objects.

Also, HyperMask is an interesting demonstration system for its integrated component technologies. Basically, HyperMask consists of camera that observes the stage, and a retro-projector that projects image information (e.g. onto the masks of the actors). Note that the retro-projector can be considered as a camera whose direction of propagation of light is inverted. Our first technical step was to implicitly calibrate the geometry implied by the camera and projector without explicitly calculating all intrinsic and extrinsic parameters, which is time-consuming and error-prone.

Another technology is real-time lip synchronization using user's own texture mapping. This system allows the user to quickly fit a face texture to a 3D polygonal model. Then, a neural network is trained for predicting lip movements based on vowels. The system can then synchronize the lip movements of the face model with the voice of the user in real time. The expression of the projected face can also be altered by the user.

The HyperMask project improves technologically some seminal work explored in different ways by contemporary artists. Projecting human faces and bodies onto mostly static objects and puppets has appeared in many art pieces, notably in the works of Tony Oursler and Laurie Anderson in the 1990s. Before that, some artists have also experimented with video-based heads for performers. In particular, Otavio Donasci has explored "video-creatures" since the beginning of the 1980s, in performances where actors used video monitors as their heads. The actors' faces were hidden from the public and substituted for the faces of off-stage actors captured live by a video camera.

2 HyperMask prototype

The first prototype of the HyperMask was used in a performance at the Sony Computer Science Labo-

1

2

Fig. 1. Installed camera and projector
Fig. 2. HyperMask prototype

ratory in the summer of 1998. After that, we incorporated the lip-synching and facial expression control software described in this paper. Based on these early experiments, we created a performance piece for the SIGGRAPH'99 Emerging Technologies exhibition that used a portable version of the HyperMask system. The equipment (camera, projector and computer) is loaded into a trolley, and the actor wheels the trolley around the performance area and chats with the audience. The faces projected onto the mask reflect the tone and content of the various stories and interactions.

Figures 1 and 2 show this HyperMask prototype. The camera on the trolley is always tracking the actor's mask, and the LCD projector is always projecting a synthesized facial expression onto the mask. The actor's speech, picked up through a microphone in the mask, is converted into a lip shape in real-time, and the lip shape image is generated. Then a face image, with a facial expression chosen by the user via a small keypad, is synthesized using a 3D face model and texture mapping. The actor can also change the face model and texture using the keypad.

3 Camera and projector calibration

A major goal of the HyperMask project was to allow the actor using the mask to move her face, so she can use facial gestures, look to the audience, nod, etc. To accomplish this we made the projected face considerably smaller than the projectable area, so if the actor moves around the projector's cone of light, it is possible to keep the computer-graphics (CG) face projected on her face by simply moving the CG face around the projectable area.

In the HyperMask system a camera is employed to track the position of the mask by finding on the camera imagery the position of 4 infrared markers on the mask. To project the CG face exactly on the mask worn by the actor, it is necessary to calibrate the camera to the projector, i.e., to determine for any point in the camera image its corresponding point in the projector's image.

The relationship between points observed on a planar surface from two different cameras is known to be a homography (Faugeras 1993). A homography (also called collineation, since it preserves lines) is a 3×3 matrix defining a linear application in the projective space that, for a given planar surface of the real world, maps all projected points in one camera's image into the other camera's image.

The fundamental observation is that from a geometrical point of view, "ideal" pinhole projectors and cameras are identical (see Fig. 3). Let $\boldsymbol{H}$ denote the homography that relates the image of the projector image frame to the camera image frame. This means that a 2D point homogeneous coordinates on the camera image,

$$\boldsymbol{c} = (x_c/z_c, y_c/z_c) ,$$

matches a 2D point,

$$\boldsymbol{p} = (x_p/z_p, y_p/z_p) ,$$

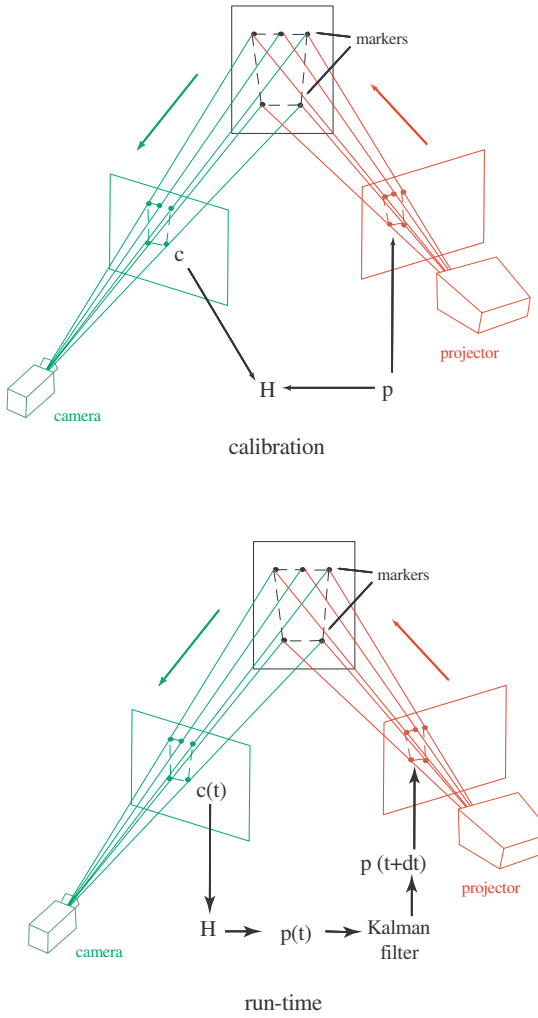


Fig. 3. Calibration process and run-time system

on the projector image as follows:

$$\boldsymbol{p} = \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} = \boldsymbol{H}c = \boldsymbol{H} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} .$$

A homography is completely defined if the projection of four 3D points of the world on both image planes is known. To determine the homography between a camera and a projector, we need simply to obtain the four needed points while manually aligning a projection of the surface with the real surface (see Fig. 3).

The homogeneous coordinates of four points to be projected,

$$p_i = (x_p^i, y_p^i, 1) \qquad i = 1, 2, 3, 4 ,$$

are determined arbitrarily, making sure that the points are visible and there is a way to move the real surface so it aligns with the projection. Then, we consider the homogeneous coordinates of the four points on the camera image as sensed by the tracking system,

$$c_i = (x_c^i, y_c^i, 1) \qquad i = 1, 2, 3, 4 .$$

Let the homography matrix, $\boldsymbol{H}$, be defined as follows:

$$\boldsymbol{H} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} .$$

Matrix $\boldsymbol{H}$ is defined up to a scalar coefficient. Assuming $h_9 \neq 0$, we set $h_9 \neq 1$.
A point with homogeneous coordinates (xyw) is transformed to $(x''y''w'')$ as below:

$$\begin{aligned} x'' &= h_1 x + h_2 y + h_3 w , \\ y'' &= h_4 x + h_5 y + h_6 w , \\ w'' &= h_7 x + h_8 y + h_9 w . \end{aligned}$$

Setting $w = 1$ yields

$$\begin{aligned} x' &= \frac{x''}{w''} = \frac{h_1 x + h_2 y + h_3}{h_7 x + h_8 y + 1} , \\ y' &= \frac{y''}{w''} = \frac{h_4 x + h_5 y + h_6}{h_7 x + h_8 y + 1} . \end{aligned}$$

This can be written as

$$\begin{aligned} x' &= h_1 x + h_2 y + h_3 - h_7 x x' - h_8 y x' , \\ y' &= h_4 x + h_5 y + h_6 - h_7 x y' - h_8 y y' . \end{aligned}$$

We obtain the following linear system to solve (we need to invert the 8×8 matrix):

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1'x_1 & -x_1'y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y_1'x_1 & -y_1'y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2'x_2 & -x_2'y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y_2'x_2 & -y_2'y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3'x_3 & -x_3'y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y_3'x_3 & -y_3'y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4'x_4 & -x_4'y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y_4'x_4 & -y_4'y_4 \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \end{pmatrix} = \begin{pmatrix} x_1' \\ y_1' \\ x_2' \\ y_2' \\ x_3' \\ y_3' \\ x_4' \\ y_4' \end{pmatrix} ,$$

where $x'_i = x'^i_p$, $y'_i = y'^i_p$ and $x'_i = x'^i_c$, $y'_i = y'^i_c$ for $i \in \{1, 2, 3, 4\}$.
Computing homographies is quite an unstable numerical process. Indeed we need to invert a 8×8 matrix. Therefore, we may use singular value decompositions or pseudo-inverse if $\boldsymbol{H}$ is ill-conditioned. Another alternative is not to use points but corners, as suggested in Zoghlami et al. (1997). A corner is defined by two half-lines joining in an intersection junction. If more fiducials are available, we can compute more reliably the homography by using least median square methods or even better the statistical approach of Kanatani (1998).
However, if the plane is far enough from both the projector and camera (in relation to their baseline distance), we can relax the homography by an affine transform (or even similitude) as described below. In fact, our final system used this simplified approach:
Taking the matrices corresponding to these two sets of four points,

$$\boldsymbol{P} = \left(p_1^{\mathrm{T}}, p_2^{\mathrm{T}}, p_3^{\mathrm{T}}, p_4^{\mathrm{T}}\right)$$

and

$$\boldsymbol{C} = \left(c_1^{\mathrm{T}}, c_2^{\mathrm{T}}, c_3^{\mathrm{T}}, c_4^{\mathrm{T}}\right) \ ,$$

we want $\boldsymbol{P} = \boldsymbol{HC}$, whose solution is

$$\boldsymbol{H} = \boldsymbol{P}\boldsymbol{C}^{\mathrm{T}} \left(\boldsymbol{C}\boldsymbol{C}^{\mathrm{T}}\right)^{-1} \ .$$

During run-time, we simply take a point in camera image $c = (x_c, y_c, 1)$, project it through the homography $\boldsymbol{H}$, obtain $p = \boldsymbol{H}c$ and compute the position on the projector's image plane,

$$\boldsymbol{p} = (x_p/z_p, y_p/z_p) \ .$$

Surprisingly, this calibration step is numerically stable even with only four points, and can be done, in practice, in a few seconds. We believe that the stability is also related to the fact that in our experiments the projection centers of the camera and the projector are close to being aligned. Note that there is no need to determine the camera's intrinsic parameters or those of the projector.

4 Tracking the projection surface

In our experiment, we used plain markers on the projection surface. In particular, we employed infrared LEDs that can be easily tracked by a camera with an infrared filter. However, if we move the mask too quickly, we observe that the projected image "falls behind" the moving surface. That is, there is a "shifting" effect, where the observations at discrete time t on the camera image, $c(t)$, are displayed by the projector at time $t + \mathrm{d}t$ using the estimate at time t, $p(t) = \boldsymbol{HC}(t)$. To reduce the "shifting" problem we employ a predictive Kalman filter (Gelb 1974) that estimates the most likely position of every point at time $t + \mathrm{d}t$, using equations of dynamics as the underlying model of the Kalman filter, as shown in Fig. 3. The parameter $\mathrm{d}t$, corresponding to the average delay between sensing and displaying, is determined experimentally. The Kalman filtering approach proved to be very effective in our experiments.

5 Handling a 3D mask

The method described above works quite well for a planar surface. However, when transferring from a 2D mask to a 3D mask, we have to handle the projected pattern more carefully. Given our projection setup, two kinds of problems occur. First, when the mask is panned to the left or to the right, hidden (or occluded) parts of the virtual projected mask do not appear on the physical mask. One ideal solution is to have a set of cameras and projectors covering the whole stage. Each projector would have to project an image on the parts of the mask it can effectively hit through a ray emanating from its optical center. However, in a performance situation it is possible to constrain the interaction with the audience so almost always the actor is looking forward, so projection occlusion is minimized.
The second problem arising from projection onto a 3D mask is related to the fact that the projection has to be corrected for the differences in depth in the projected surface. For example, suppose we have a mask in the shape of a human face, onto which a "clown" mask with a red nose is projected. Now suppose we are projecting a 2D CG rendition of a face. Also assume that the tracking system is able to correctly detect the borders of the mask and to deform the CG face to match the borders of the mask. As we rotate the mask from center to left, the projection of the corrected 2D face will, in general, put the clown's red nose in the incorrect place. This is because the nose, when viewed in a profile, moves more to the right than the rest of the face, simply because its 3D position is in front of the other parts of the face.

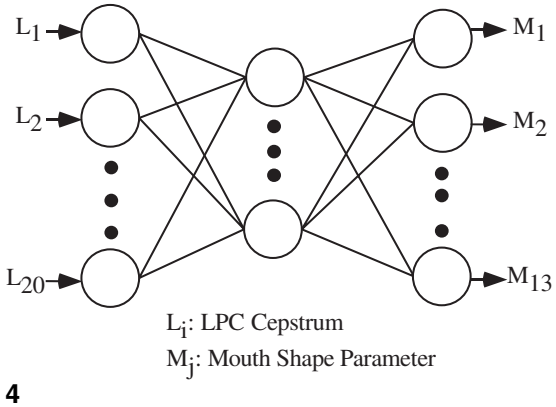
The simple way to correct for this is to project a 3D CG model of a face so it replicates in the virtual world the movements of the 3D mask in the real world. Therefore, we have to recover the latitude (Lowe 1991), i.e. the 3D coordinates in the frame world, of our 3D mask, in order to put its model in a virtual 3D scene so that we can perform occlusion and project the observed 3D scene (a 2D image) onto the mask in the 3D world. Our experiment exhibits these occluding problems. Image quality can be enhanced by using standard technique such as splatting and deghosting.

6 Real-time talking head

To realize real-time lip synchronization, the user's voice (captured by a microphone) is phonetically analyzed and converted to a mouth shape and expression parameters on a frame-by-frame basis. LPC Cepstrum parameters are converted into mouth shape parameters by a neural network trained on vowel features. Figure 4 shows the neural network structure for parameter conversion. The 20-dimensional Cepstrum parameters are calculated every 32 ms with a 32 ms frame length. The mouth shape is then synthesized according to these mouth-shape parameters. The facial expression is chosen by the user, from Anger, Happiness, Disgust, Surprise, Fear and Sadness. Each basic emotion is associated with specific facial expression parameters described by FACS (Ekman and Friesen 1978).

7 Designing the mouth shape

The set of mouth shapes can be easily edited by our mouth-shape editor (see Fig. 5). We can change each mouth parameter to determine a specific mouth shape, which can be seen in the preview window. Typical vowel mouth shapes are shown in Fig. 6. Our special mouth model has polygons for the teeth and the inside of the mouth. A tongue model is now under construction. When converting from the LPC Cepstrum parameters to the mouth shape, only the mouth shapes for 5 vowels and nasals are defined in the training set. We have defined all of the mouth shapes for Japanese phonemes and English phonemes using this mouth-shape editor.



4

5

Fig. 4. Network for parameter conversion from voice to mouth shape
Fig. 5. Mouth shape editor

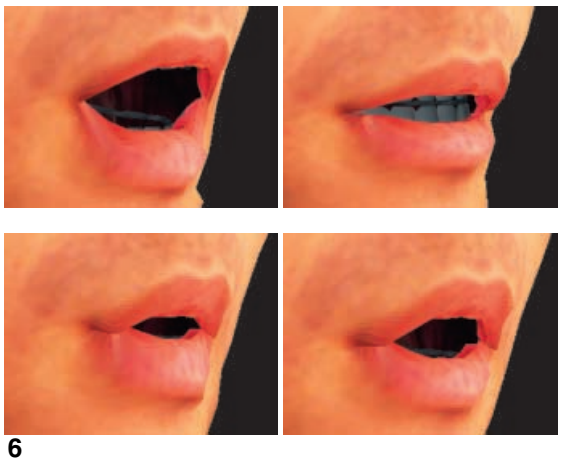
8 Customizing the face model

To generate a realistic avatar's face, a generic face model is manually adjusted to the user's face image. To produce a personal 3D face model, both the user's frontal face image and profile image are necessary. The generic face model represents all of the control rules for facial expressions (de-

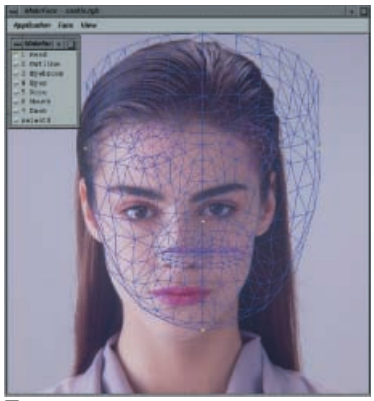
fined by FACS parameters) as a 3D movement of grid points, which modify the geometry of the model.
Figure 7 shows a personal model both before and after the fitting process for a front-view image, using our original GUI-based face-fitting tool. The front-view image and the profile image are loaded into the system, and then the corresponding control points are manually moved to an approximately correct position, using the mouse. The synthesized face results from mapping a blended texture (generated from the user's frontal image and profile image) onto the modified personal face model.
However, sometimes self-occlusion happens, and we cannot capture the whole texture using only the front and profile face images. To construct the 3D model more accurately, we introduce a multi-view, face-image-fitting tool. Figure 8 shows the fitted result with face images from any oblique angle. The rotation angle of the face model can be controlled in the GUI preview window to achieve the best fit for face images captured from any arbitrary angle. Figure 9 shows examples of reconstructed faces. Figure 9a uses 9 view images, and Fig. 9b uses only frontal and profile views. As you can see, much better image quality is achieved by the multi-view fitting process.

9 User adaptation of voice

When a new user comes in, the voice model, as well as the face model, has to be registered before operation. Ideally, the neural network has to be retrained in each case. However, it takes a very long time to get convergence using back-propagation. So, 75 subjects' voice data, including 5 vowels, were pre-captured, and a database of weights of the neural network and the voice parameters were constructed. So, speaker adaptation is performed by choosing the optimum weights from the database. When a new non-registered speaker comes in, s/he has to speak 5 vowels into a microphone. The LPC Cepstrum is calculated for the 5 vowels, and this is fed into the neural network. The mouth shape is then calculated by selected the weight, and the error between the true mouth shape and the generated mouth shape is calculated. This process is applied to all of the database entries one by one, and the optimum weight is selected when the minimum error is detected.



6



7a

7b

Fig. 6. Typical mouth shapes

Fig. 7a,b. Frontal model fitting using the GUI tool. **a** Initial model; **b** fitted model

8

9a

9b

10

Fig. 8. Multi-view fitting to oblique angle

Fig. 9a,b. Reconstructed face. **a** Multi-image view; **b** two-image view

Fig. 10. Projected face on mask

10 Interactive experience

In our proposed performance, the user is an actor portraying a storytelling character (see Fig. 10). During the stories, the attendees are the audience at a live computer-assisted performance. Between stories, however, they can chat with the character. The actor can improvise because the combination of real-time lip synchronization, active projection, and user-controlled facial expressions does away with the need for a fixed script. Surprisingly, coordinating the story-telling process with the man-

ual control of the facial expressions took the actors only about 1 h. After that period, it became quite natural to produce any desired facial expression by clicking the corresponding button on the keypad.
The HyperMask system uses an SGI Indigo2 workstation (MIPS 10000, 123 MB, IRIX6.5), a camera (Sony EVI-G20), an LCD projector (Sony), and a LED-marked mask. The chambermaid costume, wig, shopping cart, and linen are optional. A scene of live demo is shown in Fig. 10. This demonstration was made in the SIGGRAPH'99 Emerging Technology exhibition area. Hundreds of people watched the stories and interact with the two performers behind the mask (Kim Binsted and Claudio Pinhanez). Normally, 5 to 10 people at a time gathered around the performance.

11 Future vision of HyperMask

The HyperMask system is a combination of different technologies, and each will have different social, cultural and technical implications. Active projection could be useful in a number of different applications. For example, in the so-called "Office of the Future", we would like to be able to project dynamically images and information onto moving, irregularly shaped objects. We plan to extend the system to use several cameras and projectors, so that objects can be covered with projected images, which can then be viewed from any direction. More interestingly, we are also considering the use of a system with one fixed projector whose image is deflected by a rotating mirror, similar to the "everywhere displays projector" proposed by Pinhanez (2001). We also hope to be able to make the object markers more subtle, or even remove the need for them completely.
Talking heads with real-time lip synchronization also have a number of potential applications, most obviously as avatars for virtual communities and gaming. We also like to imagine people being able to put themselves into famous movies, by substituting their face for Harrison Ford's (Morishima 1996) (see Fig. 11). In addition, we have proposed "Danger Hamster 2000" (Binsted et al. 2000), which is an entertainment system that uses the HyperMask's technologies (see Fig. 12).
Computer-enhanced live performance in general shows a lot of promise. In order to support human performers in their task of entertaining and interacting with a live audience, the technology needs to be flexible, fast, and provide new creative opportunities. We believe that HyperMask is a first step in this direction.

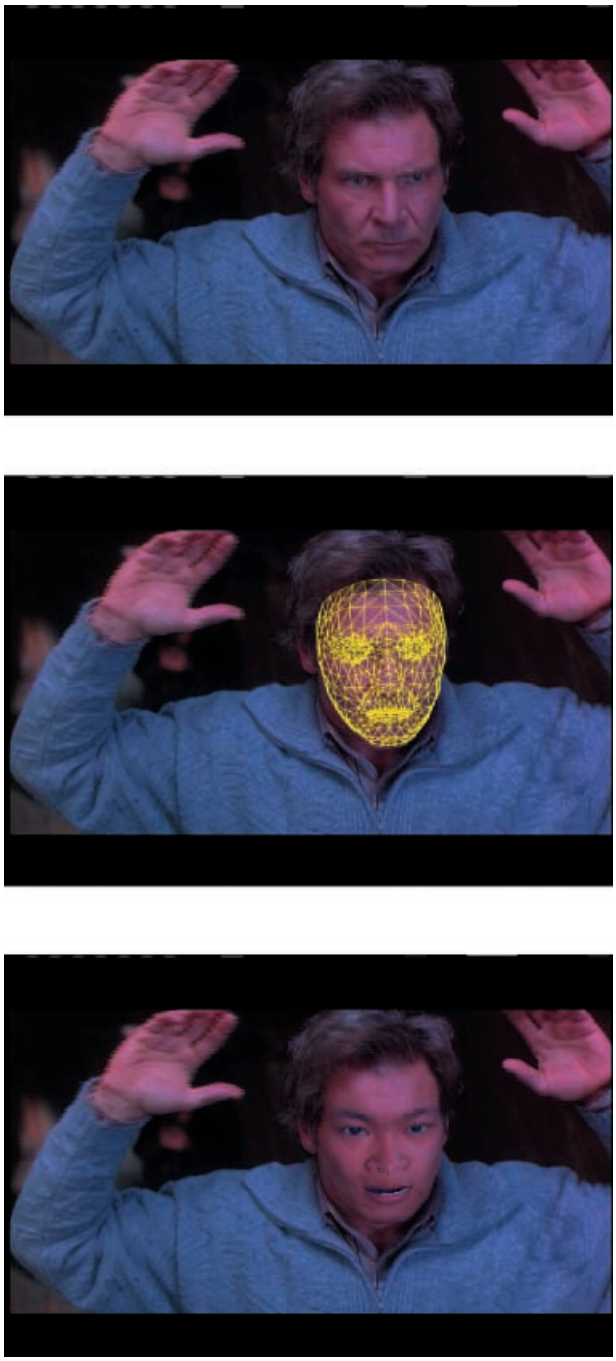


Fig. 11. Interactive movie

Fig. 12. Danger Hamster 2000 system

12 Conclusion

We have described HyperMask, a system for projecting images onto an actor's mask as that mask moves around in a performance area. The projected image is an animated face with real-time lip synchronization with the actor's voice. The face's expression is controlled by the actor to fit with the tone and content of the story being told. We also described the HyperMask prototype system, which was put into a linen cart pushed around by a chambermaid, a character who tells amusing stories and chats with the audience.

References

1. Rasker R, Welch G, Cutts M, Lake A, Stesin L, Fuchs H (1998) The office of the future: a unified approach to image-based modeling and spatially immersive displays. In: SIGGRAPH Annual Conference Proceedings. ACM, New York, pp 179–188
2. Faugeras O (1993) Three-dimensional computer vision: a geometric viewpoint. The MIT Press, Cambridge, Mass.
3. Gelb A (1974) Applied optimal estimation. The MIT Press, Cambridge, Mass.
4. Lowe DG (1991) Fitting parameterized three-dimensional models to images. IEEE Trans PAMI 13(5):441–450
5. Ekman P, Friesen WV (1978) Facial action coding system. Consult Psychol 13(5):441–450
6. Morishima S (1996) Modeling of facial expression and emotion for human communication system. Displays 17:15–25
7. Kanatani K (1998) Optimal homography computation with a reliability measure. In: Proceedings of MVA'98, IAPR Workshop on Machine Vision Applications CE[a]. pp 426–429
8. Zoghlami I, Faugeras O, Deriche R (1997) Using geometric corners to build a 2D mosaic from a set of images. In: Proc IEEE Computer Vision and Pattern Recognition. IEEE, Piscataway, NJ, pp 420–425
9. Pinhanez C (2001) Using a steerable projector to and a camera to transform surfaces into interactive displays. In: CHI'01 Conf Proc – Short Talks, Seattle. ACM, New York
10. Binsted K, Nielsen F, Morishima S, Misawa T (2000) Danger Hamster 2000. In: ACM SIGGRAPH Conference Abstracts and Applications. ACM, New York, pp 81

Photographs of the authors and their biographies are given on the next page.

CE[a] Please add publisher's name and city location

TATSUO YOTSUKURA received his B.S. and M.S. degrees, both in the Faculty of Engineering, from Seikei University, Tokyo, in 1998 and 2000, respectively. Currently, he is a Ph.D. student at Seikei University and an intern researcher at the ATR Media Integration & Communication Laboratories. His research interests include facial animation and realtime face-to-face communication systems. He is a member of IEICE-J. He received the NICOGRAPH/MULTIMEDIA best paper award in 2000 and the IEICE-J young engineer award.

SHIGEO MORISHIMA received his B.S., M.S. and Ph.D. degrees all in electrical engineering from the University of Tokyo in 1982, 1984, and 1987, respectively. Currently, he is a professor at Seikei University, Tokyo. His research interests include physics-based modeling of face and body, facial-expression recognition and synthesis, human–computer interaction, and future interactive entertainment using speech and image processing. He was a visiting researcher at the University of Toronto from 1994 to 1995. He has been engaged in the Multimedia Ambiance Communication TAO research project as a sub-leader since 1997. He has been a temporary lecturer at Meiji University, Japan, since 2000 and a visiting researcher at the ATR Spoken Language Translation Research Laboratories since 2001. He is an editor of Transactions of the Institute of Electronics, Information and Communication Engineers, Japan (IEICE-J). He received the IEICE-J achievement award in May 1992.

FRANK NIELSEN received his B.S. and M.S. degrees from École Normale Supérieure (ENS) at Lyon in 1992 and 1994, respectively. He defended his Ph.D. thesis on "Adaptive Computational Geometry" prepared at INRIA Sophia-Antipolis under the supervision of Pr. Boissonnat in 1996. As a civil servant of the University of Nice (France), he gave lectures at the engineering schools ESSI and ISIA (École des Mines). In 1997, he served in the army as a scientific member of the computer-science laboratory of École Polytechnique (LIX). In 1998, he joined Sony Computer Science Laboratories, Tokyo, as an associate researcher. His current research interests include computational geometry, algorithmic vision, combinatorial optimization for geometric scenes and compression.

KIM BINSTED is CEO of I-Chara Inc., a Tokyo-based mobile agent company (www.i-chara.com). Formerly, she was a researcher at the Sony Computer Science Laboratories, working on Human Computer Interaction and Artificial Intelligence (AI). She received her Ph.D. in AI at the University of Edinburgh and her B.Sc. in physics at McGill University, Montreal.

CLAUDIO PINHANEZ received his B.S. degree in mathematics and his M.S. degree in computer science from the University of Sao Paulo in 1985 and 1989, respectively. He received his Ph.D. from the Media Arts & Sciences Graduate Program at the Media Laboratory, Massachusetts Institute of Technology. Currently, he is research scientist at the IBM T.J. Watson Research in New York. His interests are interactive spaces; user models for human–computer interaction; machine social, intelligence, computerized entertainment; interactive stories; and computer theater.