HYPER MASK – Projecting Virtual Face on Moving Real Object

S.Morishima, T.Yotsukura, F.Nielsen [†] , K.Binsted [‡] and C.Pinhanez [§]

Faculty of Engineering, Seikei University, Tokyo, Japan

Abstract

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 propose to 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.*

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" or an interactive playground, we could 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. on the masks of the actors). Notice 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. Facial expression can also be manipulated by the user.

2. HYPERMASK Prototype

We envision a performance piece which uses a portable version of the HYPERMASK system. The equipment (camera, projector and computer) would be loaded into a trolly, and the actor would wheel the trolley around the performance area and chat with the audience. The faces projected onto the mask would reflect the tone and content of the various stories and interactions.

Figure 1: *Installed Camera and Projector*

[†] Sony Computer Science Laboratories, Tokyo, Japan
[‡] i-chara, Tokyo, Japan
[§] IBM T.J. Watson Research, Hawthorne, New York, U.S.A.

Figure 2: HYPERMASK *Prototype*

Figure 1 and Figure 2 are showing a HYPERMASK prototype. Camera on a trolly is always tracking facial mask of actor and LCD projector is always projecting a synthesized facial expression onto the mask. All of the voice spoken by actor is converted lip shape on real time process and synthesized lip shape is generated. Then face image with facial expression decided by user's manipulation is synthesized by modifying 3d model and texture mapping. Of course, actor can change face model and texture by himself scene by scene.

3. Camera and Projector Calibration

The relationship between points observed on a planar surface from two different cameras is known to be a homography [1]. A homography is a 3 by 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 Figure 3). Let 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

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

matches a 2d point

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

on the projector image as follows:

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

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 projector, we

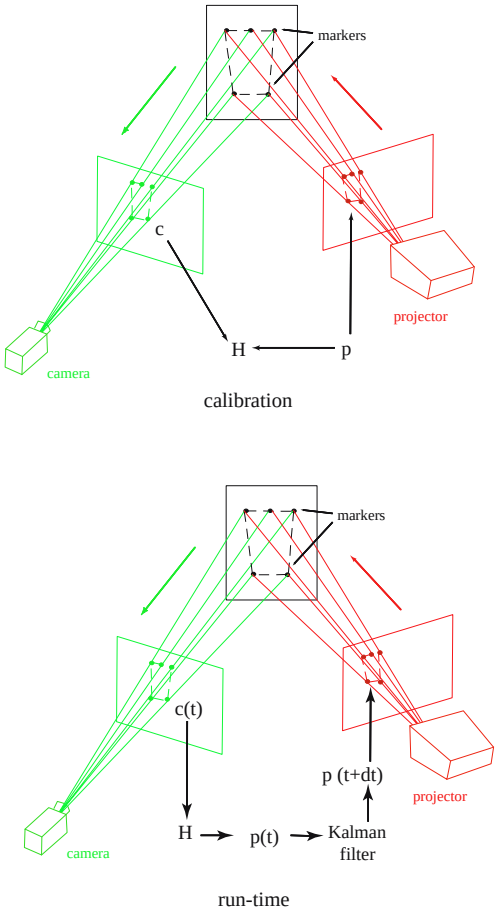


Figure 3: *Calibration process and run-time system*

need simply to obtain the four needed points while manually aligning a projection of the surface with the real surface (see Figure 3).

The homogeneous coordinates of four points to be projected,

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

are determined arbitrarily, although 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,

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

Taking the matrices corresponding to these two sets of four points,

$$P = (p_1^T, p_2^T, p_3^T, p_4^T)$$

and

$$C = (c_1^T, c_2^T, c_3^T, c_4^T)$$

we obtain $P = HC$, whose solution is

$$H = PC^T(CC^T)^{-1}$$

During run-time, we simply take a point in camera image $c = (xc, yc, 1)$, project through the homography H obtaining $p = Hc$ and compute the position on the projector's image plane,

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

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 be aligned. Notice that there is no need to determine neither the camera's intrinsic parameters nor the projector's.

4. Tracking the Projection Surface

In our experiment, we have 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 "fall 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 $t+dt$ using the estimate at time t, $p(t) = Hc(t)$. To reduce the "shifting" problem we employ a predictive Kalman filter [2], that estimates the most likely position of every point at time $t+dt$, using equations of dynamics as the underlying model of the Kalman filter, as shown in Figure 3. The parameter dt, 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

When going from a 2d mask to a 3d mask, we have to handle the projected pattern more carefully, since hidden (or occluded) parts of the virtual projected mask should 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. For example, suppose we have a mask consisting of a big nose in relief glued to a planar filled polygon. As we rotate the mask from left to right, part of the nose will not be seen by the camera or hit by the projected light. Therefore, we have to recover the latitude [3], 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 like splatting and deghosting.

6. Real-time Talking Head

To realize lip synchronization, user's voice captured by microphone is phonetically analyzed and converted to mouth shape and expression parameters frame by frame basis. LPC Cepstrum parameters are converted into mouth shape parameters by neural network trained by vowel features. Figure 4 shows neural network structure for parameter conversion. 20 dimensional Cepstrum parameter are calculated every 32ms with 32ms frame length. And then mouth shape is synthesized by this mouth shape parameters. Facial expression is manipulated by user into one of Anger, Happiness, Disgust, Surprise, Fear and Sadness. Each basic emotion has a specific facial expression parameters described by FACS [4].

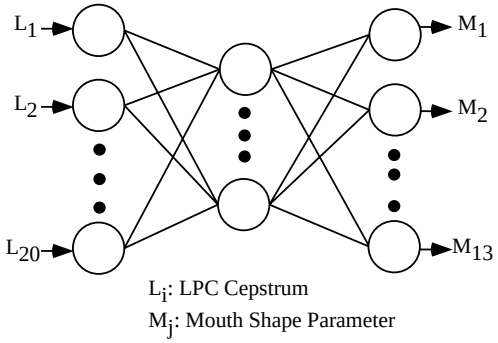


Figure 4: *Network for Parameter Conversion from Voice to Mouth Shape*

7. Designing Mouth Shape

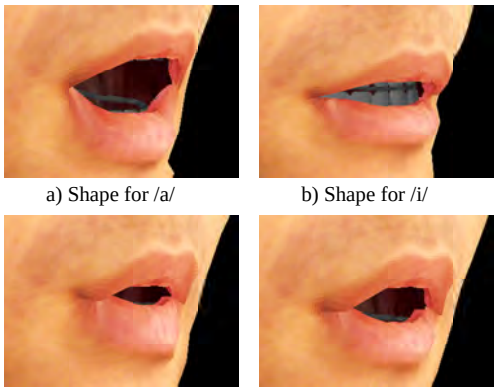
Mouth shape can be easily edited by our mouth shape editor (see Figure 5). We can change each mouth parameter to decide a specific mouth shape on preview window. Typical vowel mouth shapes are shown in Figure 6. Our special mouth model has polygons for mouth inside and teeth. Tongue model is now under construction. For parameter conversion from LPC Cepstrum to mouth shape, only mouth shapes for 5 vowel and nasals are defined as training set. We have defined all of the mouth shapes for Japanese phoneme and English phoneme by using this mouth shape editor.

8. Customizing Face Model

To generate a realistic avatar's face, a generic face model is manually adjusted to user's face image. To produce a personal 3D face model, both user's frontal face image and profile image are necessary at least. The generic face model has all of the control rules for facial expressions defined by FACS parameter as a 3D movement of grid points to modify geometry.

Figure 7 shows a personal model both before and after fitting process for front view image by using our original GUI based face fitting tool. Front view image and profile

Figure 5: *Mouth shape editor*



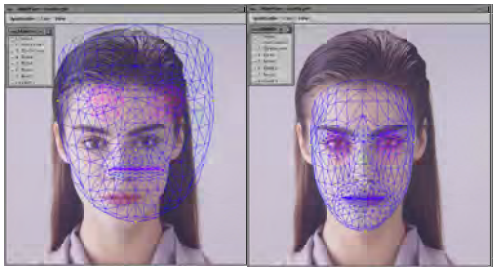
a) Shape for /a/ b) Shape for /i/

c) Shape for /u/ d) Shape for /o/

Figure 6: *Typical mouth shapes*

image are given into the system and then corresponding control points are manually moved to a reasonable position by mouse operation. Synthesized face is coming out by mapping of blended texture generated by user's frontal image and profile image onto the modified personal face model.

However, sometimes self-occlusion happens and then we cannot capture texture only from front and profile face image in the occluded part of face model. And also to construct 3D model more accurately, we introduce multi-view face image fitting tool. Figure 8 shows the fitting result with face image from any oblique angles. Rotation angle of face model can be controlled in GUI preview window to achieve best fitting to face image captured from any arbitrary angle. Figure 9 shows examples of reconstructed faces. Figure 9(a) is



a) Initial Model b) Fitted Model

Figure 7: *Frontal model fitting by GUI tool*

Figure 8: *Multi-view fitting to oblique angle*

using 9 views images and Figure 9(b) is using only frontal and profile views. Much better image quality is achieved by multi-view fitting process.

9. User Adaptation of Voice

When new user comes in, voice model has to be registered before operation as well as face model. New learning for neural network has to be performed ideally in each case. However, it takes a very long time to get convergence of backpropagation. So, 75 persons' voice data including 5 vowels are pre-captured and database for weights of neural network and voice parameters are constructed. So speaker adaptation is performed by choosing the optimum weight from database. Training of neural network for every 75 persons' data has already finished before operation. When new nonregistered speaker comes in, he has to speak 5 vowels into microphone. LPC Cepstrum is calculated for every 5 vowels and this is given into the neural network. And then mouth shape is calculated by selected weight and error be-

a) Multi-view b) Two-view

Figure 9: *Reconstructed face*

Figure 10: *Projected face on mask*

tween true mouth shape and generated mouth shape is evaluated. This process is applied to all of the database one by one and the optimum weight is selected when the minimum error is detected.

10. Interactive Experience

In our proposed performance, the user is an actor portraying a storytelling character(see Figure 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.

The HYPERMASK system uses a SGI Indigo2 workstation (MIPS 10000, 123MB, IRIX6.5), a camera (Sony EVI-G20), a LCD projector (Sony), and a LED-marked mask. Chambermaid costume, a wig. a shopping cart, and some linen are optional. A scene of live demo is shown in Fig.10. This demonstration was held at SIGGRAPH?99 Emerging Technology. 20-30 people at a time could gather round 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, so-called "Office of 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. 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 [5] (see Figure 11).

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.

12. Conclusion

We have described about HYPERMASK, a system for projecting images onto an actor's mask as that mask moves around in the 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 proposed HYPERMASK prototype system put into linen cart pushed around by chambermaid, a character who tells amusing stories and chats with the audience.

References

1. O.Faugeras, "Three-Deimensional Computer Vision: A Geometric Viewpoint", *The MIT Press*, Cambridge, Massachusetts, (1993).
2. A.Gelb, "Applied Optimal Estimation", *The MIT Press*, Cambridge, Massachusetts, (1974).
3. D.G.Lowe, "Fitting Parameterized Three-Dimensional Models to Images", *IEEE Trans. on PAMI*, Vol.13, No.5, pp.441-450, (1991).

4. P.Ekman and W.V.Friesen, “Facial Action Coding System”, *Consulting Psychologists Press Inc.*, Vol.13, No.5, pp.441-450, (1978).

5. S.Morishima, “Modeling of Facial Expression and Emotion for Human Communication System”, *Displays 17*, pp.15-25, Elsevier, (1996).

Original movie clip

Fitting result of face model

Reconstructed face by user's

Figure 11: *Interactive movie*