M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 442
Appears: Proceedings of ICCV'98, Bombay, India, January 4-7, 1998

3D Modeling and Tracking of Human Lip Motions

Sumit Basu, Nuria Oliver, and Alex Pentland
MIT Media Laboratory, 20 Ames St., Cambridge, MA 02139 USA
{sbasu,nuria,sandy}@media.mit.edu

Abstract

We address the problem of tracking and reconstructing 3D human lip motions from a 2D view. This problem is challenging due both to the complex nature of lip motions and the minimal data available from a raw video stream of the face. We counter both of these difficulties with statistical approaches. We first build a physically-based 3D model of lips and train it to cover only the subspace of lip motions. We then track this model in video by finding the shape within the subspace that maximizes the posterior probability of the model given the observed features. In this study, the features are the likelihoods of the lip and non-lip color classes: we iteratively derive forces from these values to apply to the physical model and converge to the final solution. Because of the full 3D nature of the model, this framework allows us to track the lips from any head pose. In addition, because of the constraints imposed by the learned subspace of the model, we are able to accurately estimate the full 3D lip shape from the 2D view.

1 Introduction

The lips are a critical factor in human communication: they are an important cue for speech recognition and are among the primary components of facial expressions. As a result, the detailed shape of the lips is an important input for systems that wish to observe, process, and code human communication. However, obtaining this input has proved to be quite difficult. When we are presented with a speaker's lips in a video stream, there is not a lot of visual information to work with. This is especially true given the complex ways in which lips can move. Furthermore, while the head is typically moving about in 3D, we have available only the 2D projection. Even the contours of the lips are often obscured by the lighting, facial hair, or other disturbances. Among the only features we can depend on is the color content of the lips and surrounding regions, though even this information is often noisy.

Our goal is to find a means of robustly estimating the 3D lip shape despite these limitations. It has been clear from the beginning of our work that in order to reach this goal, we would need a 3D model so that the lips could be tracked from any pose. In addition, we realized that this model would have to conform to a strict subspace that would contain only the permissible lip shapes. Without these constraints on the shape, the model could never estimate the full 3D shape from only noisy 2D information. Lastly, we would like this model to be physically based so that we can apply the standard tools of deformation and equilibrium to drive it from local observations. To satisfy these needs, we have constructed an FEM model of the lips and trained it with 3D data to conform to the subspace of lip motions.

Even with this model, we are left with the formidable challenge of tracking the lips in video data. Again we choose a statistical approach, modeling the color distributions of the lip and non-lip regions. We are then able to compute likelihood maps of each class over the relevant region of the input stream. By then projecting our 3D model into the 2D camera view, we can measure the posterior probability of the model shape using this information. This method allows us to find the correct lip pose by matching the distributions of the observations and those implied by the model, thereby maximizing the posterior probability of the model shape given the observations.

In this paper, we will describe in detail how we have formulated each of these techniques and how we apply them to this problem. We will then show our results, demonstrating how our statistical framework has allowed to us accurately and robustly estimate and reconstruct the 3D lip shape from 2D video data.

1.1 Background

In looking at the prior work on lip modeling, there are two major groups of models. The first of these contains the models developed for analysis, usually intended for input into a combined audio-visual speech recognition system. The underlying assumption behind most of these models is that the head will be viewed from only one known pose. As a result, these models are only two-dimensional. Many are based directly on image data [5],[7]; others use such low level features to form a parametrized description of the lip shape [1].

Some of the most interesting work done in this area has been in using a statistically trained model of lip variations. Bregler and Omohundro's work and Luettin's work, for example ([4] and [9]), model the subspace of lip bitmaps and contours respectively. However, since these are 2D models, the changes in the apparent lip shape due to rigid rotations have to be modeled as complex changes in the lip pose. One goal of this paper is to extend these ideas to 3D. By modeling the true three-dimensional nature of the lips, variations that look complex and nonlinear from a 2D perspective become simple and linear. With a 3D model, we can simply rotate the model to match the observed pose, modeling only the actual variations in lip shape.

The other category of lip models includes those designed for synthesis and facial animation. These lip models are usually part of a larger facial animation system, and the lips themselves often have a limited repertoire of motions [8]. To their credit, these models are mostly in 3D. For many of the models, though, the control parameters are defined by hand. A few are based on the actual physics of the lips: they attempt to model the physical material and musculature in the mouth region [6],[12]. Unfortunately, the musculature of the mouth is extremely complicated and has proved to be very difficult to model accurately. Even if the modeling were accurate, this approach would still

result in a difficult control problem. Humans do not have independent control of all of these facial muscles: normal motions are a slim subspace of the possible muscle states. Some models have tried to approximate this subspace by modeling key lip positions (visemes) and then interpolating between them (for example [12]). However, this limits the accuracy of the resulting lip shapes, since only the key positions are learned from data.

We hope to fill the gap in these approaches with a 3D model that can be used for both analysis and synthesis. Our approach is to start with a 3D shape model and generic physics. We then deform this initial model with real 3D data to learn the correct modes of variation, i.e., all of the deformation modes that occur in the observations. In this way, we not only address the problem of parametrizing the model's motions, but also that of control. Because we learn only the modes that are observed, we end up with degrees of freedom that correspond only to plausible motions.

2 The Model

In the following section, we give a brief description of the choice of model shape and the physics used. A much more detailed account is given in [2].

The underlying representation of our initial model is a mesh in the shape of the lips. The mesh is constructed from a linear elastic material modeled by the Finite Element Method (FEM). The initial shape for the model was obtained by extracting a region surrounding the lips from a Viewpoint Data Labs model of the human head. Some small changes were made in this initial model to make it suitable for the finite element framework.

The FEM is a numerical method for approximating the physics of an arbitrarily complex body. The individual stress-strain matrices of the elements can be assembled into a single, overall matrix expressing the static equilibrium equation

$$\mathbf{KU} = \mathbf{F} \tag{1}$$

where the displacements $\mathbf{U}$ and forces $\mathbf{F}$ are in a global coordinate system. The details of this method are described in many references (e.g., [3]).

For this application, a thin-shell model was chosen. We constructed the model by beginning with a 2D plane-stress isotropic material formulation and adding a strain relationship for the out-of-plane components (see [2] for further details).

It is important here to understand the difference between a physically-based and a physiological model. We are not attempting to construct a physiological model, and thus we do not claim that our model has any simple relation to the actual stiffnesses of the skin, muscle, and other tissue that make up the mouth region. Our model is a thin shell structure, while the actual lips are clearly volumetric in nature. What we do claim is that our model (after training) can accurately account for the visible *observations* of the mouth. The "learned physics" that we discuss here corresponds to learning the modes and distributions of deformations that account for these observations. The framework of the physical model is simply a means of modeling these observations that allows us to conveniently model the interrelations between different parts of the structure.

3 The Observations

To train this model to have the correct 3D variations of the lips, it was necessary to have accurate 3D data. Also, in order to observe natural motions, it was not acceptable to affix reflective markers or other cumbersome objects to the lips. To satisfy these criteria, seventeen points were marked on the face with ink: sixteen on the lips and one on the nose. The placement of these points is shown in figure 1. The points were chosen to obtain a maximally informative sampling of the 3D motions of the lips.

Figure 1: Locations of marked points on the face

Once the points were marked, two views of the points were taken by using a camera-mirror setup to ensure perfect synchronization between the two views. The points were tracked over 150 frames at a 30Hz frame rate using supervised normalized correlation. The two views were then used to reconstruct the 3D location of the points. Finally, the points were transformed into a head-aligned coordinate system to prevent the rigid motion of the head from aliasing with the non-rigid motions of the lips.

It was attempted to have as great a variety of lip motions within this brief period as possible. To this end, several utterances using all of the English vowels and the major fricative positions were spoken during the tracking period. Clearly, 150 frames from one subject is still not enough to cover all possible lip motions, but it is enough to provide the model with the initial training necessary to cover a significant subset of motions. Methods to continue the training using other forms of input data will be discussed in a later section.

4 Training the Model

In order to relate the training data to the model, the correspondence between data points and model nodes had to be defined. This was a simple process of examining a video frame containing the marked points and finding the nodes on the lip model that best matched them in a *structural* sense. The difference between the goal locations of these points (i.e., the observed point locations) and their current location in the model is then the displacement goal, $\mathbf{Ug}$.

4.1 Reaching the Displacement Goals

The issue was then how to reach these displacement goals. The recorded data points constrained 48 degrees of freedom (16 points on the lips with three degrees of freedom each). However, the other 564 degrees of freedom were left open. To solve this underconstrained problem, we added the constraint of minimum strain. Given the set of constrained point displacements, our solution minimized the strain felt throughout the structure. This solution is thus a physically based smoothing operation: we are using the physics of the model to smooth out the regions where we have no observation data by minimizing the strain in the model. We denote the the $\mathbf{K^{-1}}$ matrix with only the rows pertaining to the constrained degrees of freedom as $\mathbf{P}$. The minimum strain solution can then be expressed as:

$$\hat{\mathbf{F}} = \mathbf{P^T}(\mathbf{PP^T})^{-1}\mathbf{Ug} \tag{2}$$

Details of the derivation can be found in [2].

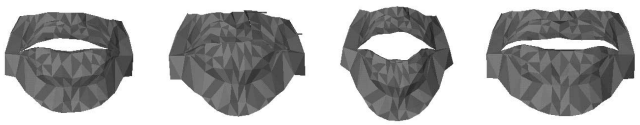

Figure 2: The mean displacement and some characteristic modes

4.2 Modeling the Observations

Once we have all the displacements for all of the frames, we can relate the observed deformations to a subset of the "correct" physics of the model. We began with the default physics (i.e., fairly uniform stiffness, only adjacent nodes connected) and have now observed how the model actually deforms. This new information can be used to form a new, "learned" $\mathbf{K}$ matrix. Martin *et al.* [10] described the connection between the strain matrix and the covariance of the displacements $\mathbf{R_u}$: if we consider the components of the force to be IID with unit variance, we have

$$\mathbf{R_u} = \mathbf{K_s^{-2}} \tag{3}$$

We can now take this mapping in the opposite direction. Given the sample covariance matrix $\hat{\mathbf{R}}_\mathbf{u}$, we can find $\mathbf{K^{-1}}$ by taking its positive definite square root, i.e., diagonalizing the matrix into $\mathbf{S\Lambda S^T}$ (where each column of $\mathbf{S}$ is an eigenvector and $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues) and then reforming it with the square roots of the eigenvalues. We can then use the resulting "sample $\mathbf{K^{-1}}$" to represent the learned physics from the observations. Forces can now be applied to this matrix to calculate the most likely displacement given the observations.

However, because we only have a small number of training observations (140) and a large number of degrees of freedom (612), we could at best observe 140 independent degrees of freedom. Furthermore, noise in the observations makes it unreasonable to estimate even this many modes. We thus take only the 10 linear modes that account for the greatest amount of variance in the input data (i.e., those with the largest eigenvalues of the covariance matrix). These modes are found by performing principal components analysis (PCA) on the sample covariance matrix. We can then reconstruct the modal covariance and $\mathbf{K^{-1}}$ matrices using these modes. We thus have a parametric description of the subspace of lip shapes (the modes) and a probability measure for the subspace (the modal covariance matrix).

The sample covariance was computed using only the first 140 frames so that the last ten could be used as a test set. The mean displacement ($\bar{\mathbf{U}}$) and some of the first few modes are shown in figure 2 below. It was found that the first ten modes cover 99.2 percent of the variance in the data. We should thus be able to reconstruct most shape variations from these modes alone.

5 Tracking the Lips in Raw Video

Now that we have the model, we turn to the task of tracking the model in raw video. By "raw" we mean that there are no longer any special markings on the lips or the face. We have only the minimal features described in the introduction. Among these, the color content of the various regions is a robust and easily computable candidate. However, it will not directly give us any kind of shape information - it will only give us the likelihoods of membership in the color classes, $l_{model} = f(color|model)$. If we view the problem from a statistical perspective, it becomes clear how this data should be used. Essentially, we want to find the set of parameters p^* for our model that maximizes its posterior probability given the observations:

$$p^* = \arg\max_p f(p|O)) = \arg\max_p \frac{f(O|p)(f(p))}{f(O)} \tag{4}$$

we can neglect the denominator in the last expression, since it will be the same for all p, leaving us with

$$p^* = \arg\max_p f(O|p)(f(p)) \tag{5}$$

Another piece of information we have is the color class of each point on our model. As shown in the figures above, the model contains the lips and some surrounding skin, and we know a priori which triangular faces belong to which class. If we now project the model in state p into the camera view, we can compute the term $f(O(x,y)|p)$ for each point in the visible surface of the model. This value is simply the likelihood of the observed color value at (x,y) belonging to the same class as the point in the model that is projected onto it. To find the overall probability of the model in this state, we simply integrate these values over the visible surface area A and postmultiply by the prior value of the model being in state p:

$$f(p|O) = \int_A f(O(x,y))|p)f(p) \tag{6}$$

This gives us a measure of the posterior probability of the model being in a given state. We will show in a later section how this integral can be decomposed for efficient computation. This still leaves the problem of finding the optimal state without searching the entire subspace. We approach this using a form of stochastic gradient ascent that makes use of the physical basis of the model. Based on the likelihoods and the gradients thereof, we derive forces to "push" the model state in a direction that will increase its overall likelihood.

In order to apply these ideas to our tracking problem, we first train models of the color classes for the skin and lips. We then detect and find the face within the image frame. Next, we find the lips within the face and compute the detailed likelihood maps for the skin and lips classes. The model is then positioned to match the coarse statistics of the lip distribution projected into the camera view. From this initial fit, we iteratively derive forces from the observations as described above and apply them to the model. We then measure the probability of the resulting states and stop the algorithm when we reach a local maximum. We will describe each of these steps in detail in the following sections.

5.1 Training the Color Classes

We derive the statistical models of the lip and skin (face) classes using the LAFTER system [11], a real-time active-camera face tracking system. This system uses examples of lip and skin pixels to build models of the probability distributions of each class in color space. The distributions are modeled as mixtures of Gaussians and are estimated using the EM algorithm.

Past studies ([13]) have shown that use of normalized or chromatic color information ($\tilde{r} = \frac{r}{r+g+b}, \tilde{g} = \frac{g}{r+g+b}$) can be reliably used for finding "flesh areas" present in the scene despite wide variations in lighting. By training the model on this normalized space on thousands of skin color samples, we have obtained a model that is valid for a broad spectrum of users.

Typically, only two to three mixture components are needed to accurately characterize the face. Lip models are even more densely distributed in the chromatic space since they have a

considerably smaller color variance. One or two mixture components (in color space) are typically sufficient to accurately describe lips.

Once we have these models, we can apply them to the relevant regions of the image to produce *probability maps* for the lip and the skin classes. The accurate modeling of the skin and lip classes result in sharp boundaries between high and low probability regions for each class (see figures 3 and 4). As a result, the probability maps have a very high gradient at the boundary and very low gradients everywhere else. Because we wish to derive forces from the gradients to push the model towards the maximally probable point in the subspace, we need to spread their region of influence to a larger area. To do this, we convolve the probability maps with a 2D Hamming window, resulting in smoothed probability maps (see figures 3 and 4). In addition, the values of the raw probability maps in the highly probable region tend to have high variance, resulting in fairly noisy gradients. The smoothing counteracts this problem as well. Figures 3 and 4 show the original and smoothed likelihoods of lips ($l_{lips}(x,y)$) and skin ($l_{skin}(x,y)$) for a typical input image.

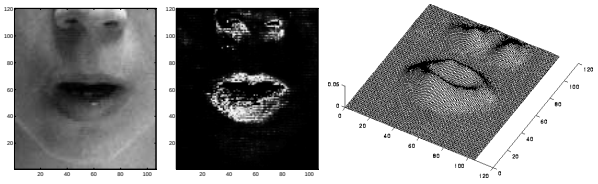

Figure 3: Original image, lip PDF, and smoothed lip PDF

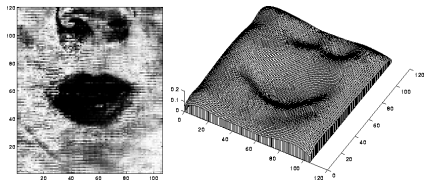

Figure 4: Skin PDF and smoothed skin PDF

5.2 Detecting and Finding the Face and Lips

To estimate the location of the face and the lips in the image, the LAFTER system makes use of 2-D *blob features*, spatially-compact clusters of pixels that are statistically similar in terms of low-level image properties. See [11] for the motivation and history of blob features.

In our implementation, feature vectors are computed at each pixel by concatenating the (x,y) spatial coordinates and the color components at that point. These features are then clustered so that image properties such as color and spatial similarity combine to form coherent connected regions, or "blobs," in which all the pixels have similar image properties.

Once the pixels are clustered into blobs, the largest blob of skin-class pixels is tested for size. If it is of sufficient size, the blob is considered a face. The probability model for the lip color is then applied to the pixels in this face blob along with the prior model for lip location within the face. The resulting lip blob with the highest a posteriori probability is then taken to be the subject's lips. The location of the lips (i.e., the mean of the blob) is then tracked from frame to frame with a Kalman filter.

5.3 Projecting the Model into the Camera View

The rigid pose of the model is related to the camera view by six rigid parameters: three for rotation and three for translation. The 2D projection of the model into the camera view is found using a pinhole camera model with a calibrated focal length. The location of the lip blob gives us enough information to estimate the in-plane translation automatically (in the near future, we plan to have the out-of-plane rotation and depth parameters determined automatically by a visual head-tracking system). To do this, we match the centroid of the visible area of the lip region of the model to the centroid of the blob estimate. This is a first-order alignment of the observed statistics and the statistics implied by the model. From this initial fit, we can iteratively deform the model to maximize the probability of the model state given the observations.

5.4 Measuring the Model Probability

In order to measure the probability of the current model state given the observations, we need to now compute the expression in equation 6, which integrates the probability of each point over the visible area of the model. We can break this expression up into a sum of integrals over the faces (the triangular facets) of the model:

$$f(p|O) = f(p) \sum_i \int_{face_i} f(O(x,y)|p) \tag{7}$$

Furthermore, we can evaluate the integral over each face using Gaussian numerical integration with a single sample point [3]. In the one point case, the approximation to the integral of a function over a triangular patch is the area of the patch A_i multiplied by the value at the center of the patch $f(O(\bar{x}_i,\bar{y}_i)|p)$. Note that this scheme approximates the function as being constant over the face. This is reasonable for our situation because of the small size of the faces (and thus the small variation in the function surface over them - see figure 5). The resulting approximation to the total integral is:

$$f(p|O) = f(p) \sum_i A_i f(O(\bar{x}_i,\bar{y}_i))|p) \tag{8}$$

We can now simplify $f(O(x,y)|p)$ to $l_{lips}(x,y)$ or $l_{skin}(x,y)$, depending on whether the given face in the model is a lip face or a skin face. This thus breaks the result into two pieces:

$$\begin{aligned} f(p|O) &= f(p) \sum_{lipfaces} A_i l_{lips}(\bar{x}_i,\bar{y}_i) \\ &+ f(p) \sum_{skinfaces} A_i l_{skin}(\bar{x}_i,\bar{y}_i) \end{aligned} \tag{9}$$

Lastly, to evaluate the prior term $f(p)$, we use a Gaussian model with the learned covariance $\mathbf{Ru}_{,modal}$. Because we already have the parameters p in the modal coordinate system, the exponent term can be simplified to the sum of ten terms (for the ten modes).

$$\sum_{i=1}^{10} \frac{f_i^2}{\lambda_i} \tag{10}$$

The scaling factor for the Gaussian can be omitted since it is the same for all values of p.

With these simplifications, the final expression in equation 9 can be quickly computed from the available quantities.

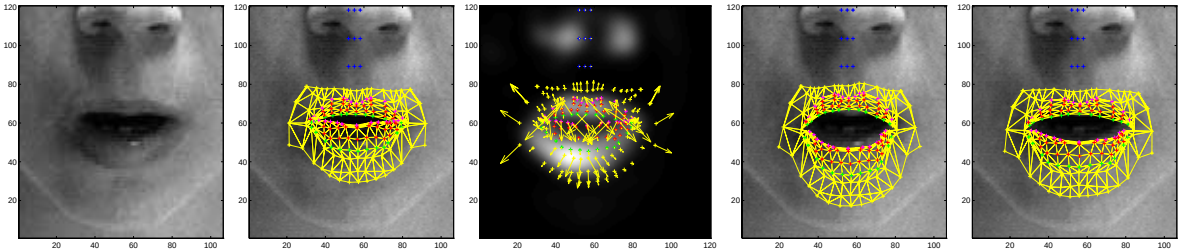

Figure 5: From initial image to final fit

5.5 Iterating to a Solution

Now that we can compute the posterior probability of the model pose, we need a means of improving it. The method we have developed is a form of stochastic gradient ascent. For each face, we find the local direction in which its likelihood would increase the most (i.e., the gradient of the appropriate probability map). We then compute a force proportional to the gradient to apply to the face pushing it in this direction. Mathematically, we want to integrate the influence of the gradients over the entire face:

$$f_i = \alpha \int_{face_i} \nabla f_{class}(x, y) \tag{11}$$

where *class* is lips or skin depending on the face and α is a constant that scales the force to match the physics of the model. We numerically evaluate this integral, again using the one point Gaussian model:

$$f_i = \alpha A_i \nabla f_{class}(\bar{x}_i, \bar{y}_i) \tag{12}$$

We wish to apply this force to the center of the face, which results in spreading the force in equal portions to the three nodes of the face. When the forces for all of the faces have been computed, we transform the values back into the model's coordinate sytem. We then apply the resulting force vector to the modal $\mathbf{K_S^{-1}}$ matrix, which projects the force and the resulting displacement into the subspace learned by the model:

$$\mathbf{U} = \mathbf{S}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{S}^{\mathbf{T}}\mathbf{F} \tag{13}$$

Because of the modal form of the matrix, this is a computationally inexpensive operation.

After the force has been applied, we project the new shape into the camera view and recompute the posterior probability of the model. We then iterate the process above until the probability converges to a local maximum. The resulting estimated shape is then used as the initial shape for the next input frame.

Another approach to finding this solution would be to directly compute the gradient in the parameter space and move in the direction of steepest ascent. This is the more traditional form of gradient ascent, and is typically much more expensive than stochastic ascent. However, with a small number of parameters, this method may prove more efficient due to its more accurate estimate of the gradient. We are currently exploring this alternate technique and will report on its characteristics in a future paper.

6 Results

6.1 Tracking and Reconstruction Results

In this section, we show several examples of using the above algorithm to estimate the 3D lip pose. We begin with a detailed example (figure 5): In the first frame, we see the mouth image we are trying to fit. In the next frame, we see the initial placement of the model on the image (the center of the model was aligned with the centroid of the lip probability map). In the third frame, we see the smoothed lip probability map corresponding to the image and the forces on the model points resulting from it. In the fourth frame, we see the result of the force applied in the first iteration. In the last frame, we see the final, converged result after 15 iterations. The figures below (6 and 7) show some other frames with the initial image, the final converged fit, and the profile view of the estimated model.

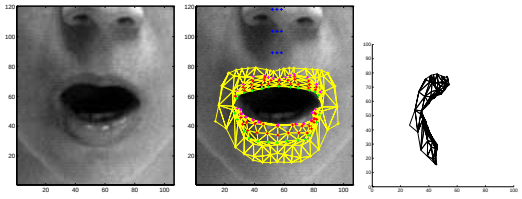

Figure 6: Initial image, final fit, and 3D reconstruction

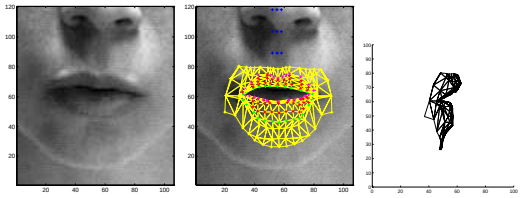

Figure 7: Initial image, final fit, and 3D reconstruction

The figures below (8 and 9) show the lip shape estimated from two partial-profile views.

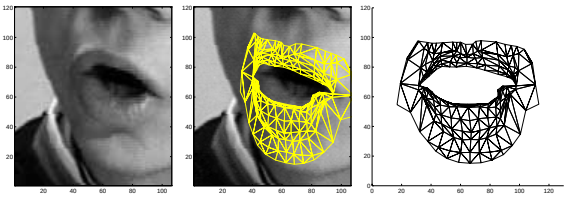

Figure 8: Initial image, final fit, and 3D reconstruction

From these images, it is clear that this algorithm can accurately estimate the 3D shape of the lips from the 2D observations.

One of the main reasons we have used a small number of modes (10) throughout this development is to be robust to

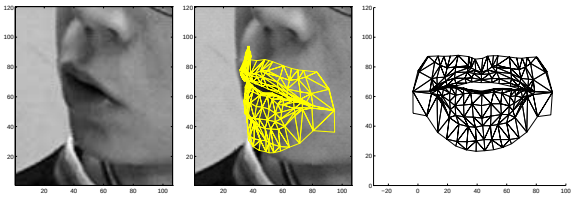

Figure 9: Initial image, final fit, and 3D reconstruction

noisy data. When clean data is available, though, we expect to be able to fit many more modes with high accuracy. This points towards a much more convenient method of continuing to train the model. Once we find the best fit in the modal space, we can relax the modal constraints and allow the initial full-rank structure to completely fit the available data. New shapes obtained in this way can then be used for additional training.

6.2 Reconstruction Capabilities

As we have previously discussed, one of the major arguments behind the 3D representation was that we could use a small number of observations from any viewpoint to find a good estimate of the model shape. We now demonstrate that we can accurately reconstruct the full shape using only y-z data, only x-y data, and x-y-z data from only a subset of the points. We show this using a test set that is separate from the training set used earlier. For the ten data frames that were not included in the sample covariance, the mean-squared reconstruction errors per degree of freedom were found for several cases and are shown in the table below. The results are given in the coordinate system of the model, in which the model is 2.35 units across, 2.83 units wide, and 5.05 units deep. The table shows the reconstruction error using only the first ten modes. The rows of the table correspond to what measurements were used to reconstruct the full 3D shape. In the first row, the first eight 3D points (shown in figure 1) were used to reconstruct the remainder of the displacements. Note that the model performs quite well in this case, implying that it has learned to some degree the permissible subset of lip motions. The second row shows the results of using only the y and z components of the data. This corresponds to the data that would be available from a profile view. The last row contains the results using the x and y components (i.e., a frontal view). It is interesting to note that the y-z data provides much better performance than the x-y case. This is understandable in that there was a significant amount of depth variation in the test frames. Because some x-y variations can occur with different degrees of z motion, the depth variation is not observable from the frontal plane. As a result, the y-z data provides more information about the lip shape in these cases. Since our model is a full 3D representation, it can take advantage of this disparity (or any other advantageous 3D pose) when these observations are available.

Data Used	3D Reconstruction Error
xyz (8 points)	1.10e-3
yz (16 points)	7.13e-4
xy (16 points)	6.70e-3

Table 1: Reconstruction error per DOF (in normalized coordinates)

7 Conclusions and Future Directions

We have presented a method for estimating and reconstructing the 3D shape of human lips from raw video data. We have shown how we can accurately match the observations in raw data, and have also demonstrated the ability of our model to accurately reconstruct 3D shapes from sparse 2D data. We have achieved these goals by using a statistical approach throughout, from modeling the subspace of lip motions to describing and fitting the observations in the video stream.

There are a number of directions in which we wish to continue this work. Foremost among these is integrating a 3D head pose estimate with the tracking algorithm so that the tracking can be robust to arbitrary changes in pose. We also plan to evaluate how much new information is provided by a 3D estimate for tasks such as audio-visual speech recognition and facial expression recognition.

8 Acknowledgements

The first author was partially supported by an NSF Graduate Research Fellowship. The second author was partially supported by La Caixa Foundation.

References

[1] A. Adjoudani and C. Benoit. "On the Integration of Auditory and Visual Parameters in an HMM-based ASR". In *NATO Advanced Study Institute: Speechreading by Man and Machine*, 1995.

[2] Sumit Basu and Alex Pentland. "A Three-Dimensional Model of Human Lip Motions Trained from Video". In *Proceedings of the IEEE Non-Rigid and Articulated Motion Workshop*, June 1997.

[3] Klaus-Jurgen Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, 1982.

[4] Christoph Bregler and Stephen M. Omohundro. "Nonlinear Image Interpolation using Manifold Learning". In *NIPS 7*, 1995.

[5] Tarcisio Coianiz, Lorenzo Torresani, and Bruno Caprile. "2D Deformable Models for Visual Speech Analysis". In *NATO Advanced Study Institute: Speechreading by Man and Machine*, 1995.

[6] Irfan A. Essa. *"Analysis, Interpretation, and Synthesis of Facial Expressions"*. PhD thesis, MIT Department of Media Arts and Sciences, 1995.

[7] Michael Kass, Andrew Witkin, and Demetri Terzopoulous. "Snakes: Active Contour Models". *International Journal of Computer Vision*, pages 321–331, 1988.

[8] Y. Lee, D. Terzopoulos, and K. Waters. "Realistic Modeling for Facial Animation". In *Proceedings of SIGGRAPH*, pages 55–62, 1995.

[9] J. Luettin, N. Thacker, and S. Beet. "Visual Speech Recognition Using Active Shape Models And Hidden Markov Models". In *ICASSP96*, pages 817–820. IEEE Signal Processing Society, 1996.

[10] John Martin, Alex Pentland, and Ron Kikinis. Shape analysis of brain structures using physical and experimental modes. In *CVPR94*. IEEE Computer Society, 1994.

[11] Nuria Oliver and Alex Pentland. Lafter: Lips and face tracking. In *CVPR97*. IEEE Computer Society, 1997.

[12] K. Waters and J. Frisbie. "A Coordinated Muscle Model for Speech Animation". In *Graphics Interface*, pages 163–170, 1995.

[13] Christopher Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.