ELSEVIER

Speech Communication 26 (1998) 131–148

SPEECH COMMUNICATION

3D lip shapes from video: A combined physical–statistical model [1]

Sumit Basu *, Nuria Oliver, Alex Pentland

Perceptual Computing Section, The MIT Media Laboratory, E15-383, 20 Ames St., Cambridge, MA 02139, USA

Received 30 January 1998; received in revised form 10 June 1998; accepted 16 July 1998

Abstract

Tracking human lips in video is an important but notoriously difficult task. To accurately recover their motions in 3D from any head pose is an even more challenging task, though still necessary for natural interactions. Our approach is to build and train 3D models of lip motion to make up for the information we cannot always observe when tracking. We use physical models as a prior and combine them with statistical models, showing how the two can be smoothly and naturally integrated into a synthesis method and a MAP estimation framework for tracking. We have found that this approach allows us to accurately and robustly track and synthesize the 3D shape of the lips from arbitrary head poses in a 2D video stream. We demonstrate this with numerical results on reconstruction accuracy, examples of static fits, and audio-visual sequences. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Lip models; Deformable/non-rigid models; Finite element models; Analysis-synthesis models; Training models from video; Model-based tracking

1. Introduction

It is well-known that lips play a significant role in spoken communication. Summerfield's classic 1979 study (Summerfield, 1979) showed how the presence of the lips alone (without tongue or teeth) raised word intelligibility in noisy conditions from 22.7% to 54% on average and up to a maximum of 71%. Not only can the lip shape be used to reduce noise and enhance intelligibilty for human/machine speech understanding, but it is also useful as a significant feature for the understanding of expression. However, to realize any of these applications, it is necessary to robustly and accurately track the lips in 3D. Why is 3D so critical? Consider the canonical task of speech recognition at a distance. By moving the microphone from the headset to the desktop and using the lip shapes to account for the new channel noise, we can free the user from cumbersome headgear. However, if he must be facing straight into the camera in order for the system to work, we have achieved only minimal freedom. In natural conversation and expression, we move our heads constantly, both in translation and rotation. If we cannot contend with this simple fact, we will never reach the unconstrained interfaces we desire. Computer vision techniques have been developed that accurately track the head's 3D rigid motions (as in our previous work (Basu et al., 1996; Jebara and Pentland, 1997)), leaving the formidable task

* Corresponding author. E-mail: sbasu@media.mit.edu.

[1] Speech files available. See http://www.elsevier.nl/locate/specom.

0167-6393/98/$ – see front matter © 1998 Elsevier Science B.V. All rights reserved.
PII: S 0 1 6 7 - 6 3 9 3 (9 8) 0 0 0 5 5 - 7

of tracking the remaining 3D non-rigid deformations.

In this paper, we develop a method for successfully facing this difficult problem. One of the most vexing issues surrounding the lip tracking problem has been the poor quality of the available data – contours, color, flow, etc., are all obscured at some point or other by lighting, the speed of motion, and so on. Our approach is thus to build and rely on strong models of the lip shape to correct for anomalies in the data. In essence, our model learns the permissible space of lip motions. The incoming data from the video stream is then *regularized* by this model – we find the *permissible* lip shape that can best account for the data. In this way, we remain robust to the unavoidable noise in the raw features. To build and train this model, we start by giving a lip-shaped mesh generic physical characteristics using the Finite Element Method (FEM). This acts as a physically based "prior" (i.e., locally elastic behavior) on how things move. We then train this model with 3D data of real lip motions and blend the physical prior with the statistical characteristics of this data. Finally, we use this physical–statistical model in a MAP estimation framework to find the locally most probable lip shape that accounts for the incoming data. Along the way, we have also developed a full-fledged synthesis model – by moving the model through the permissible lip space, we can generate images of the 3D lips in motion.

Through this method, we have been able to robustly and accurately track lip shapes in 3D from arbitrary head poses in a video stream. We will demonstrate our results with an illustration of the learned lip subspace, numerical figures on reconstruction accuracy, examples of static fits of the model, and audio-visual sequences demonstrating the tracking and synthesis in action.

1.1. Background

In looking at the prior work on lip modeling and tracking, 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 (Coianiz et al., 1995; Kass et al., 1988); others use such low level features to form a parametrized description of the lip shape (Adjoudani and Benoît, 1995).

Some of the most interesting work done in this area has been in using a statistically trained model of lip variations. Bregler and Omohundro (1995) and Luettin et al. (1996), for example, 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. In our work, we begin by extending this philosophy to 3D. By modeling the true three-dimensional nature of the lips, variations that look complex and highly nonlinear from a 2D perspective become far simpler. With a 3D model, we can 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 (Lee et al., 1995). 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 (Essa, 1995; Waters and Frisbie, 1995). 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 small 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, (Waters and Frisbie, 1995)). 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 our 3D model, which can be used for both analysis and synthesis. We start with a 3D shape model and generic physics, but 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. This yields powerful advantages for both tracking and synthesizing lip shapes. For tracking, it means we only need to search along the learned degrees of freedom (a 10-dimensional space, for example, instead of a 612-dimensional one for the unconstrained mesh), and also that we remain robust to anomalies in the data. For synthesis, it means we have a small number of "control knobs" to produce any lip shape we may need. Furthermore, as we will show, we also have a model of the probability density of the lip shape within this parametric space. We can thus trade off the likelihood of the model with the strength of the observations to find an optimally probable lip shape given the data.

2. The model

In the following section, we give a brief description of the choice of the model shape and the physics used. A more detailed account of the FEM and particulars of our implementation are provided in Appendix A.

The underlying representation of our initial model is a mesh in the shape of the lips. At the initial stage, before any training has occurred, we have no learned notion of the lip shape. We thus simply extract the region surrounding the mouth in a Viewpoint Data Labs model of the human head and made a few minor changes to aid the physical modeling steps ahead. The final model has 336 faces and 204 nodes, resulting in 612 degrees of freedom (three per node). The initial shape is shown in Fig. 1. Similarly, we have no real idea what the inherent degrees of freedom of the lips are. However, we do know something about how the lip material behaves, namely that it acts in a locally elastic way. When one portion of the lips is pulled on, the surrounding region stretches with it. We express this notion mathematically in our model by using the FEM. We use this method to give this initial mesh the properties of a generic elastic material – i.e., we treat the mesh as if it were formed from a rubber sheet. The resulting first-stage model is a "physical prior" for our training stages to come. It clearly does not have the overall correct shape modes of the lips, but when we receive 3D point locations from our training data, it tells us how to move the points in between.

2.1. More on the FEM

The FEM is a numerical technique for approximating the physics of an arbitrarily complex body by breaking it into many small pieces (elements) whose individual physics are very simple. The individual stress–strain matrices of the elements can then be assembled into a single,

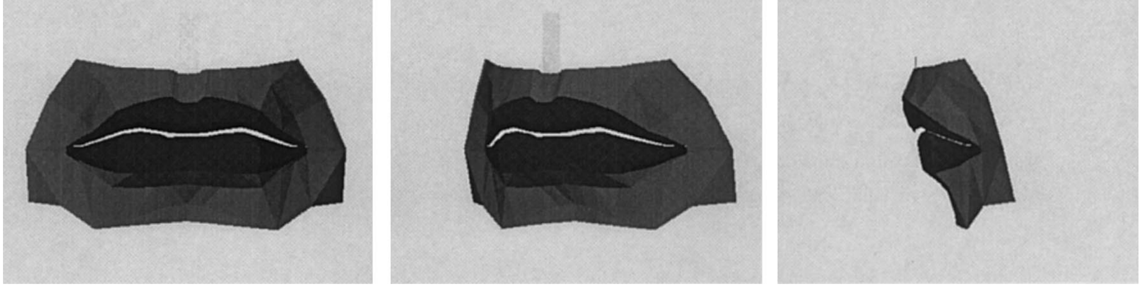

Fig. 1. Initial shape of the model: front, partial profile and full profile views.

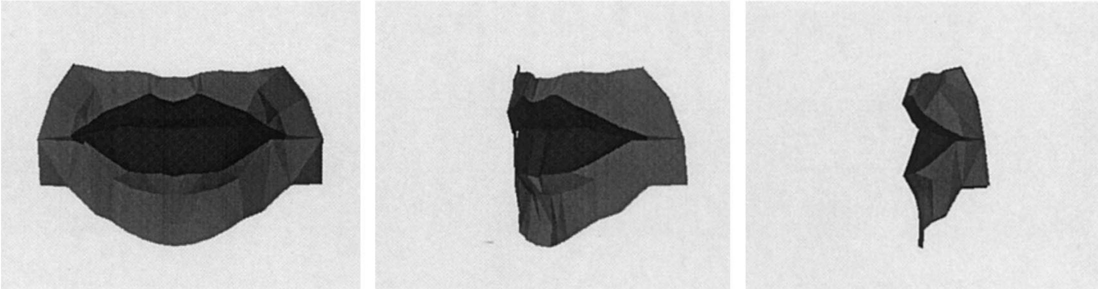

Fig. 2. Final linearization point for the model: front, partial profile and full profile views. The FEM matrix $\boldsymbol{K}$ used for all training computations was derived from this shape.

overall matrix expressing the static equilibrium equation

$$\boldsymbol{K}u = f, \tag{1}$$

where the displacements u and forces f are in a global coordinate system. The FEM thus linearizes the physics of the body around a given point. It is important to choose this nominal point carefully, as the linearization is only valid in a limited neighborhood of the point. Initially, the physics for the model were derived using a thin-shell model with the shape in Fig. 1 as a nominal point. However, because this shape is not very realistic, we used the training data and the deformation strategy described in the sections below to deform this model to a data frame that had the lips in a typical rest (closed) position. The physics were then relinearized about this point. The resulting model was the one used in all of training methods described below, and is shown in Fig. 2.

Appendix A and Basu (1997) give further details on the physical modeling techniques that were used and how they were applied to the lip model.

2.2. Understanding the meaning of the model

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 describe 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

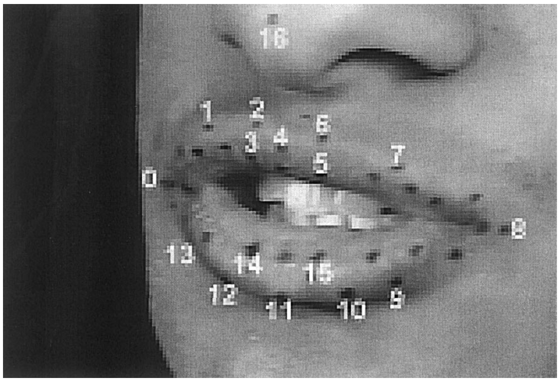

Fig. 3. Locations of marked points on the face.

and one on the nose. The placement of these points is shown in Fig. 3. The points were chosen to obtain a maximally informative sampling of the 3D motions of the lips.

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 30 Hz 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. See (Basu, 1997) for further details on these methods.

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 for continuing 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 3D point locations) and their current location in the model is then the displacement goal, u_{g}.

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. There are an infinite number of ways to reach the displacement goals – as long as the constrained points reach the goals, the other points are free to do anything. However, we wanted the physically correct solution: to pin down the constrained points and let the other points go to their equilibrium locations.

Mathematically, this idea translates to the constraint of minimum strain. Given the set of constrained point displacements, our solution must minimize the strain felt throughout the structure. This solution is thus a physically-based smoothing operation: the physics of the model are used to smooth out the regions where we have no observation data by minimizing the strain in the model.

Fortunately, in the finite element framework, this solution can be found analytically and with little computation. If we denote the $\boldsymbol{K}^{-1}$ matrix with only the rows pertaining to the constrained degrees of freedom as $\boldsymbol{P}$, the desired solution can be put in the form of the standard underconstrained least-squares problem. We wish to minimize

$$f^{\mathrm{T}}f, \tag{2}$$

with the constraint

$$\boldsymbol{P}f = u_{\mathrm{g}}, \tag{3}$$

which results in following solution (Gelb, 1974):

$$\hat{f} = \boldsymbol{P}^{\mathrm{T}}(\boldsymbol{P}\boldsymbol{P}^{\mathrm{T}})^{-1}u_{\mathrm{g}}. \tag{4}$$

If we apply this $\hat{f}$ to the mesh, we will have the desired minimum-strain displacement.

4.2. Modeling the observations

Once we have all the displacements for all of the frames, the observed deformations can be related 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 seen how the model can be deformed with point observations. The results of these deformations can now be used to form a new, "learned" $\boldsymbol{K}$ matrix. Martin et al. (1998) described the connection between the strain matrix and the covariance of the displacements $\boldsymbol{R}_{\boldsymbol{u}}$: if we consider the components of the force to be IID with unit variance, we have

$$\boldsymbol{R}_u = \boldsymbol{K}^{-2}. \tag{5}$$

We can now take this mapping in the opposite direction. Given the sample covariance matrix $\hat{\boldsymbol{R}}_u$, we can find $\boldsymbol{K}^{-1}$ by taking its positive definite square root, i.e., diagonalizing the matrix into $\boldsymbol{S}\Lambda\boldsymbol{S}^{\mathrm{T}}$ (where each column of $\boldsymbol{S}$ is an eigenvector and Λ is the diagonal matrix of eigenvalues) and then reforming it with the square roots of the eigenvalues. We can then use the resulting "sample $\boldsymbol{K}^{-1}$" 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 (150) and a large number of degrees of freedom (612), we could at best observe 150 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. The modal covariance and $\boldsymbol{K}^{-1}$ matrices can then be reconstructed 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).

Frontal, partial profile, and full profile views of the the mean displacement ($\bar{u}$) and some of the first few modes are shown in Fig. 4. Though we are only using the first ten modes, it was found that these account for 99.2% 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

At this point, we have a parametric model of the permissible lip shapes and a probability model

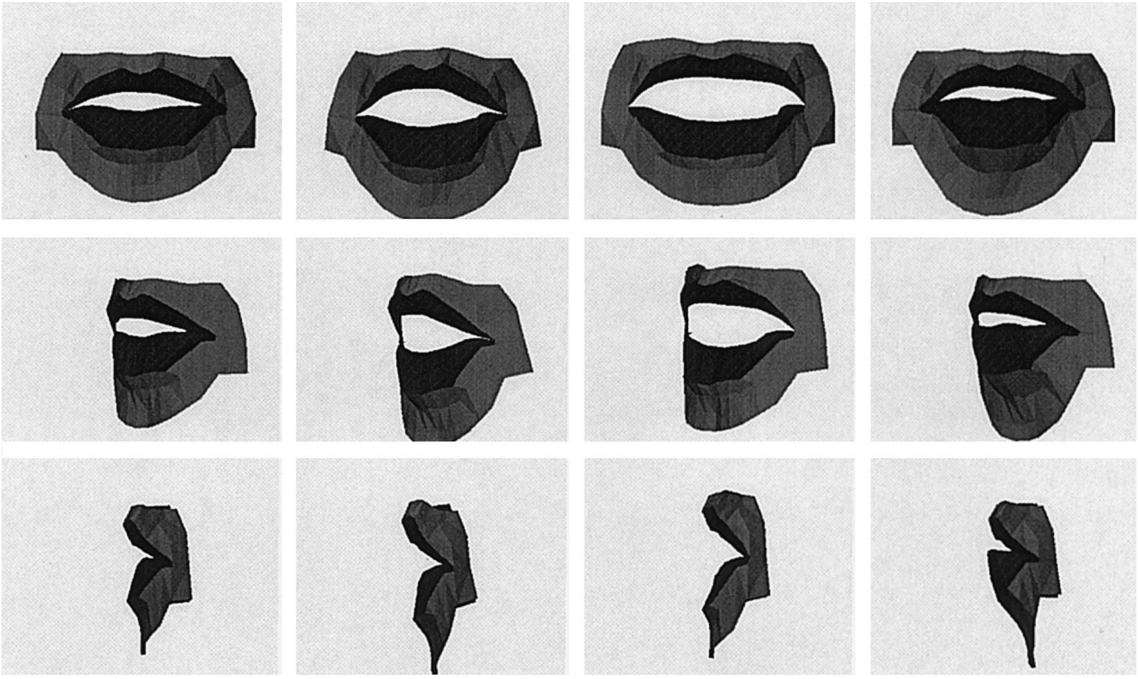

Fig. 4. Front, partial profile and full profile views of the mean displacement and some characteristic modes.

for the resulting subspace. The remaining task is to fit this model to the raw video stream in the absence of special markings on the lips or face. As mentioned earlier, our approach will be to find the lip shape within the learned subspace that best accounts for the incoming data. In statistical terms, this means finding the parameters with the highest a posteriori probability given the observations and our prior model. Intuitively, though, it simply means balancing the potentially noisy data from our observations with our learned notion of what shapes are permissible.

Any of a number of features (or a combination thereof) could be used as observations in our framework – color classification, optical flow, contours, tracked points, etc. For this implementation, we have chosen to use only the color content of the various regions, as it is a robust and easily computable candidate. Of course, this feature will not directly give us any kind of shape information – it will only give us the probabilities of each pixel belonging to the color classes, $f_{\mathrm{model}} = f(\mathrm{color}|\mathrm{model})$. From our statistical perspective, though, it is clear how this data should be used. We wish to find the set of parameters p^* for our model that maximizes its posterior probability given the observations

$$\begin{aligned} p^* &= \arg \max_p f(p|O) \\ &= \arg \max_p \frac{f(O|p)\, f(p)}{f(O)}, \end{aligned} \tag{6}$$

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

$$p^* = \arg \max_p f(O|p) f(p). \tag{7}$$

We can further simplify this by taking the logarithm. Because the logarithm is a monotonic function, maximizing the log of the expression in Eq. (7) is equivalent to maximizing the original expression

$$\begin{aligned} p^* &= \arg \max_p \log f(O|p) f(p) \\ &= \arg \max_p (\log f(O|p) + \log f(p)). \end{aligned} \tag{8}$$

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 probability 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 take the product of the observations (making an assumption of independence, which we will modify later on) over the model region and postmultiply by the prior value of the model being in state p,

$$f(p|O) = \alpha \prod_n f(O_n(x,y)|p) f(p), \tag{9}$$

where α represents a constant factor to account for $f(O)$. In the log domain, this product becomes a sum,

$$\begin{aligned} \log f(p|O) = \sum_n & \log f(O_n(x,y)|p) \\ & + \log f(p) + \log \alpha. \end{aligned} \tag{10}$$

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 quantity can be decomposed for efficient computation. This still leaves the problem of finding the optimal state without searching the entire subspace. We approach this through gradient ascent: at each step, we compute the direction in the parameter space that will most increase the fit of our model to the data. We then take a step in this direction.

In order to apply these ideas to our tracking problem, we first train models of the color classes for the skin and lips. Next, we compute the probability maps for the image (i.e., 2D maps whose entries are the probability values of the given class). The model is then initially positioned based on the rigid pose and geometry of the head. From this initial fit, we find the gradient of the optimization function in Eq. (10) in the parameter space and climb to a local maximum of the posterior probability. For the next frame, we then begin the ascent at these parameter values. We will describe this process in detail in the following sections. Note that this technique is much stabler and faster than the local-contribution method described in our earlier work (Basu et al., 1998).

5.1. Training the color classes

The statistical models of the lip and skin colors are derived by first collecting a hundred or so RGB sample points from each class for a given user. For this paper, the samples were picked by hand, but it is a simple matter to acquire them automatically, for example using a system such as (Oliver et al., 1997). The distributions are modeled as mixtures of Gaussians and are estimated using the Expectation–Maximization algorithm, using three components for the lips and one for the skin. In the past, we have modeled these classes in an intensity-normalized color space (Basu et. al, 1998), but have found that this produces less accurate features. This is especially true in dark areas where dividing by the intensity has the dangerous effect of amplifying the camera noise. Furthermore, there are subjects for whom the lip color is quite similar to the skin color, in which case it is the intensity alone which differentiates the classes. For these reasons, we now model the color classes in the full RGB space.

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. Fig. 5 shows the probability maps for the lip ($f_{\text{lips}}(x,y)$) and skin ($f_{\text{skin}}(x,y)$) classes for a typical input image.

These probability maps tend to be somewhat noisy due to camera noise, as can be seen in Fig. 5. To compensate for this, we convolve them with a 7×7 normalized Hamming kernel, resulting in smoothed probability maps. Note also that the lighting conditions can often obscure the color information available. For example, the right side of the face in Fig. 5 is too dark to provide any salient color cues. However, because we have learned the subspace of permissible lip shapes, the lip shape can be accurately estimated using only the available information (i.e., the left side of the face). The quality of the tracking results under this particular lighting condition can be viewed in the first audio-visual sequence.

5.2. Projecting the model into the camera view

The 2D projection of the model into the camera view is found using a pinhole camera model with a calibrated focal length. The rigid pose of the model is related to the camera view by six rigid parameters: three for rotation and three for translation. In addition, there are three scaling parameters (in x, y and z) that fit the lip shape to a particular user – these parameters are of course constant for a given user. For the results shown in this paper, these parameters were fit by hand in the first frame and the head was kept rigid throughout the sequence. We are currently integrating the head-tracking system in (Jebara and Pentland, 1997) to automatically determine the rigid position of the head. Given this initial rigid fit, we iteratively deform the model along the learned non-rigid modes to maximize the probability of the model state given the observations as described in the sections below.

5.3. Measuring the model probability

In order to measure the probability of the current model state given the observations, we need to

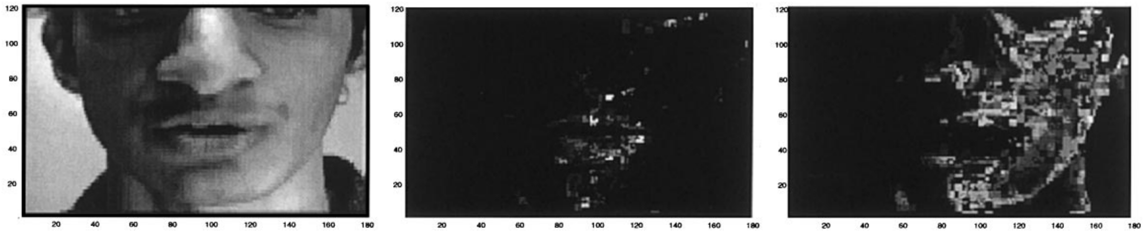

Fig. 5. Original image, lip probability map and skin probability map.

now compute the expression in Eq. (10), which sums the log probability of the observations. We can break this expression up into a sum of sums over the pixels of the faces (the triangular facets) of the model

$$\log f(p|O) = \sum_i \sum_{\text{face}_i} \log f(O(x,y)|p) + \log f(p) + \log \alpha. \quad (11)$$

Furthermore, we can reduce the computation by only sampling each face at its centroid and multiplying by the visible area of the face, which can be easily computed as the 3D area multiplied by the dot product of the view vector and the face normal (faces that are not visible do not contribute to this sum). This area A_i is thus multiplied by the value at the center of the patch, $\log f(O(\bar{x}_i,\bar{y}_i)|p)$. 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 Fig. 8). At this point, we also introduce a scaling factor γ that multiplies the resulting value for each face. By multiplying each centroid probability by the visible surface area (of A_i pixels^2), we are making the assumption that we have A_i statistically independent samples of the value at the centroid, which is clearly not the case. If we do not adjust for the mutual dependence of these samples, their contribution will overwhelm the prior ($\log f(p)$) and we would trust the data entirely. The small number of samples we are actually taking and the lack of true independence among these values does not warrant this level of trust. We thus scale all of the areas by a parameter γ which compensates for the dependence among the observations. This makes the effect of our observations commensurate with that of our prior, still giving each face a contribution proportional to its visible area.

The resulting scaled total sum is

$$\log f(p|O) = \sum_n \gamma A_n \log f(O(\bar{x}_n,\bar{y}_n)|p) + \log f(p) + \log \alpha. \quad (12)$$

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

$$\log f(p|O) = \sum_{\text{lipfaces}} \gamma A_n \log f_{\text{lips}}(\bar{x}_n,\bar{y}_n) + \sum_{\text{skinfaces}} \gamma A_n \log f_{\text{skin}}(\bar{x}_n,\bar{y}_n) + \log f(p) + \log \alpha. \quad (13)$$

Lastly, to evaluate the prior term $f(p)$, we use a Gaussian model with the learned covariance $\boldsymbol{R}_u$, modal, which takes the form

$$f(p) = \frac{1}{(2\pi)^5 * |\boldsymbol{R}_u|^{0.5}} \exp\left(-0.5 p^{\mathrm{T}} \boldsymbol{R}_u^{-1} p\right). \quad (14)$$

The log of this quantity is then simply

$$\log f(p) = C - 0.5 p^{\mathrm{T}} \boldsymbol{R}_u^{-1} p, \quad (15)$$

where C is the constant log of the scaling for the Gaussian. This can further be simplified to the sum of eleven terms, since $\boldsymbol{R}_u$ is diagonal,

$$\log f(p) = C - 0.5 \sum_{i=1}^{10} \frac{p_i^2}{\lambda_i}. \quad (16)$$

The overall posterior probability of the model can thus be written as a set of simple sums, where the new constant D is $C + \alpha$,

$$\log f(p|O) = \sum_{\text{lipfaces}} \gamma A_m \log f_{\text{lips}}(\bar{x}_m,\bar{y}_m) + \sum_{\text{skinfaces}} \gamma A_n \log f_{\text{skin}}(\bar{x}_n,\bar{y}_n) \quad (17)$$

$$- 0.5 \sum_{i=1}^{10} \frac{p_i^2}{\lambda_i} \quad (18)$$

$$+ D. \quad (19)$$

5.4. Iterating to a solution

Now that the posterior probability of the model shape can be computed, we need a means of improving it. We do this by iteratively moving in the direction that will most improve this posterior. In general, finding the explicit gradient can be quite expensive, and approximations to the gradient are used to save computation (as in our previous work (Basu et al., 1998)). However, in this case, because we have found a low-dimensional parametric space that our model is allowed to move in, we only need to compute the gradient along these dimensions. Furthermore, because our modes are linear, we do not need to relinearize our model at every step. As a result, we have found that we can compute the explicit gradients for our model very efficiently. Because this correponds to the optimal direction to move the model, it converges much more quickly and smoothly to a probabilistic maximum than our previous method.

We thus wish to find the direction of optimal ascent in our parametric space. Mathematically, we seek the quantity

$$\frac{\mathrm{d}\ \log f(p|O)}{\mathrm{d}p}. \tag{20}$$

Since $f(O)$ is constant for a given frame, we can apply Bayes' rule and then break this up as follows:

$$\frac{\mathrm{d}\ \log f(p|O)}{\mathrm{d}p} = \frac{\mathrm{d}\ \log f(O|p)}{\mathrm{d}p} + \frac{\mathrm{d}\ \log f(p)}{\mathrm{d}p}. \tag{21}$$

We will deal with these terms separately to simplify the development. The first term can be rewritten as a chain of partial derivatives,

$$\frac{\mathrm{d}\ \log f(O|p)}{\mathrm{d}p} = \frac{\partial \log f(O|p)}{\partial \bar{x}_n}\frac{\partial \bar{x}_n}{\partial p}, \tag{22}$$

which can be written more explicitly as

$$\begin{aligned}\frac{\mathrm{d}\ \log f(O|p)}{\mathrm{d}p} &= \gamma\left[A_1 \frac{\partial \log\ f(O(\bar{x}_1,\bar{y}_1)|p)}{\partial x_1} \quad \dots \quad A_N \frac{\partial \log\ f(O(\bar{x}_n,\bar{y}_n)|p)}{\partial y_n}\right] \\ &\quad \times \begin{bmatrix} \frac{\partial \bar{x}_1}{\partial p_1} & \frac{\partial \bar{x}_1}{\partial p_2} & \cdots & \cdots \\ \frac{\partial \bar{y}_1}{\partial p_1} & \frac{\partial \bar{y}_1}{\partial p_2} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \\ \vdots & \vdots & & \frac{\partial \bar{y}_N}{\partial p_M} \end{bmatrix}\end{aligned} \tag{23}$$

where N is the number of faces and M is the number of modes being used. The first term in this product is composed of the x and y derivatives of the log probability map, scaled by the visible face areas A_n and γ. The columns of the second term are simply the in-plane components of the modes computed at the centroids of each face. To find the latter, the 3D modes are rotated and scaled by the rigid transform/scaling discussed earlier. Because the modes are linear, though, for a given head pose, this second term is constant. Even when the head moves, it can be updated at minimal computation cost (the modes are simply transformed by 3 × 3 rotation/scaling matrix).

We now go on to deal with the second term in Eq. (21). The gradient of this term is

$$\frac{\mathrm{d}\ \log f(p)}{\mathrm{d}p} = \left(-\boldsymbol{R}_{\boldsymbol{u}}^{-1}p\right)^{\mathrm{T}}. \tag{24}$$

We now use these results to take a step in the direction of the overall gradient,

$$\hat{p} = \hat{p} + \beta\frac{\mathrm{d}\ \log f(p|O)}{\mathrm{d}p}, \tag{25}$$

where β is sufficiently small to account for the nonlinearities in the posterior surface in most cases. To ensure that we continue moving upwards in probability, though, the log probability is computed after each step. If it has decreased, we go back and use a smaller (half) value of β. This ascent process is continued until we have converged to a local maximum, which typically occurs in less than twenty iterations. The resulting estimated shape is then used as the initial shape for the next input frame.

6. Results

In this section, the reconstruction and tracking capabilities of our method are demonstrated. We first provide numerical results that show the capability of our model to accurately reconstruct 3D lip shapes from 2D data. Examples are then provided of using the tracking method described above to capture the lip shape from a 2D video stream and reconstruct the 3D shape. This is shown both with example fits in static frames and with audio-visual sequences. The advantages of the modal form of our model are also discussed.

6.1. Reconstruction capabilities

As noted above, one of the major arguments behind the 3D representation was that a small number of observations from any viewpoint could be used to find a good estimate of the model shape.

This is because the subspace of permissible lip shapes has been learned. Without the model, the 2D observations would leave far too many degrees of freedom unconstrained. With the model, as we will show, all degrees of freedom can accurately be reconstructed. We demonstrate this by reconstructing the 3D shape using only $x - y$ (frontal view) data and only $y - z$ (side view) data.

The mean-squared reconstruction errors per degree of freedom were found for two cases of 2D observation scenarios and are shown in Table 1. 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 0.83 units deep. The reconstruction error shown is from using only the first ten modes. Note that these results were obtained using a cross-validation method, so the errors reported are on frames outside the training set.

Table 1
Reconstruction error per DOF (in normalized coordinates)

Data used	3D reconstruction error
xy (frontal)	6.70×10^{-3}
yz (profile)	7.13×10^{-4}

The rows of the table correspond to what measurements were used to reconstruct the full 3D shape. In the first row, only the data available from a frontal view was used for the estimation. In other words, the x and y coordinates of all points were observed, but the z coordinates were not. Fig. 6 shows the dimensions of the data points used for the estimation with a '+' marker and those not used with an 'o' marker. Front and side views of the reconstruction for the point locations are also shown.

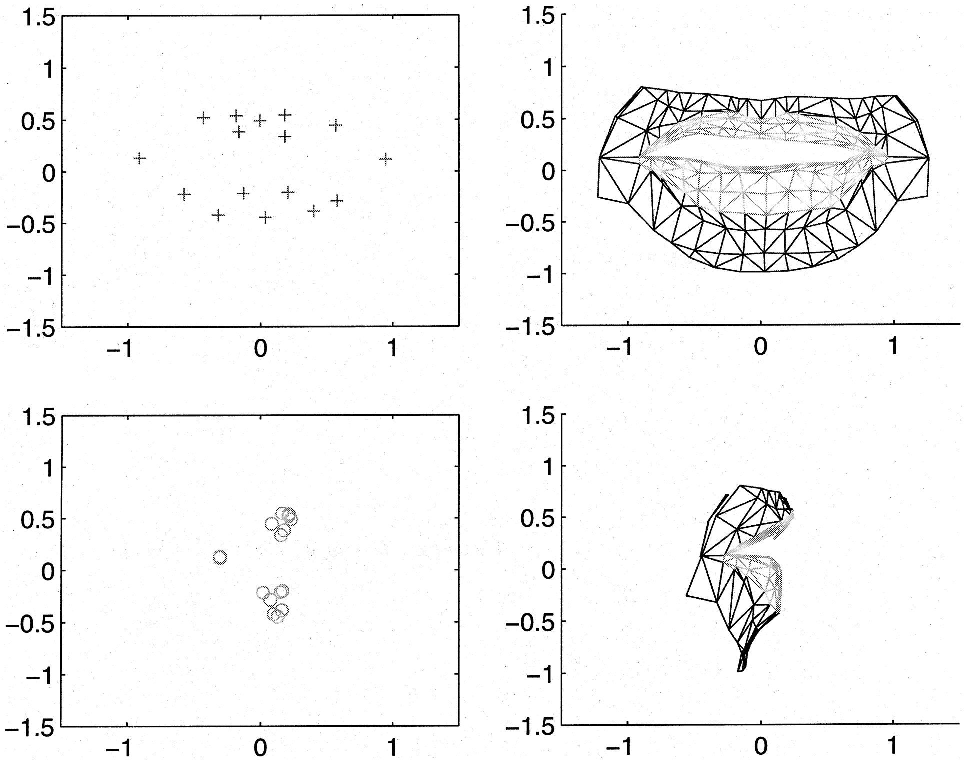

Fig. 6. Data used and a sample reconstruction for frontal view reconstruction experiment. Data point dimensions used (x and y) are labeled with a '+'; dimensions not used are labeled with an 'o'.

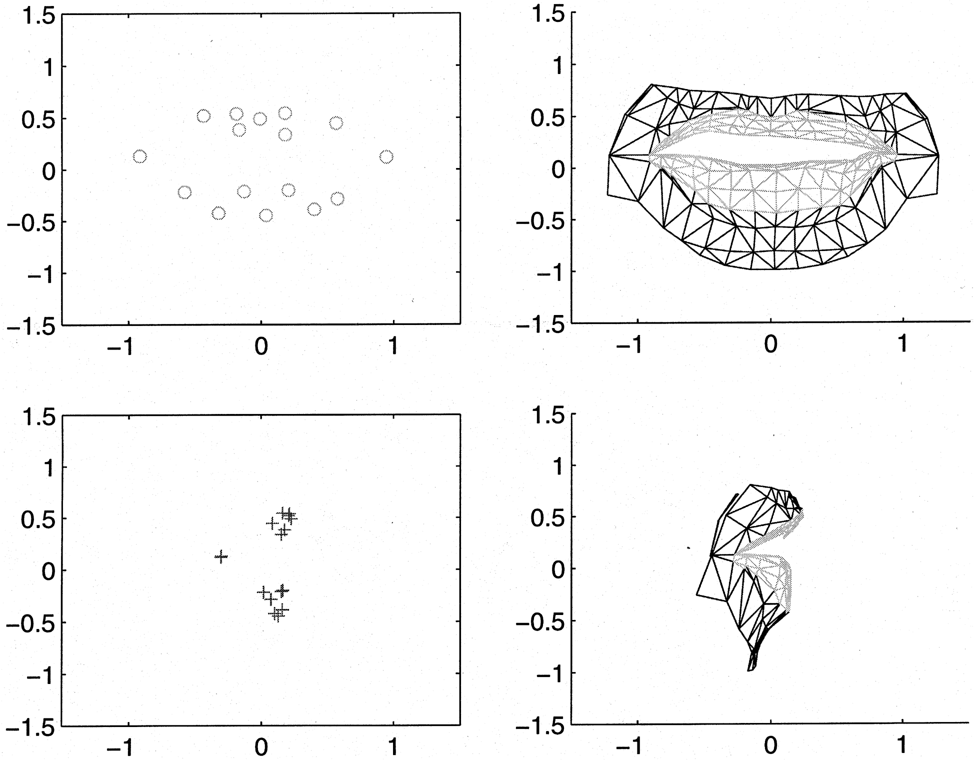

Fig. 7. Data used and a sample reconstruction for profile (side) view reconstruction experiment. Data point dimensions used (y and z) are labeled with a '+'; dimensions not used are labeled with an 'o'.

The second row shows the results of using only the data available from a profile (side) view. Here, the y and z components of the data were observed, but the x locations were not. Fig. 7 shows the coordinates used and a sample reconstruction for this experiment.

Note that in both cases (see Table 1), the reconstruction is quite accurate in terms of mean-squared error. This shows that the ten learned modes are a sufficiently strong characterization to accurately reconstruct the 3D lip shape from 2D data. 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 many $x-y$ variations can occur at a variety of z positions, 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.

6.2. Tracking and reconstruction results

In this section, several examples are provided for using our algorithm to estimate the 3D lip shape. We begin with a detailed example (Fig. 8). The first frame shows the mouth image we are trying to fit. The next frame shows the initial placement of the model on the image. The third frame shows the gradient direction resulting from the observations. In the last frame we see the final, converged result after 20 iterations.

Figs. 9–12 show some other frames with the initial image, the final converged fit, and the profile view of the estimated model. The audio-visual sequences these frames are taken from, along with the tracking and reconstruction views, are

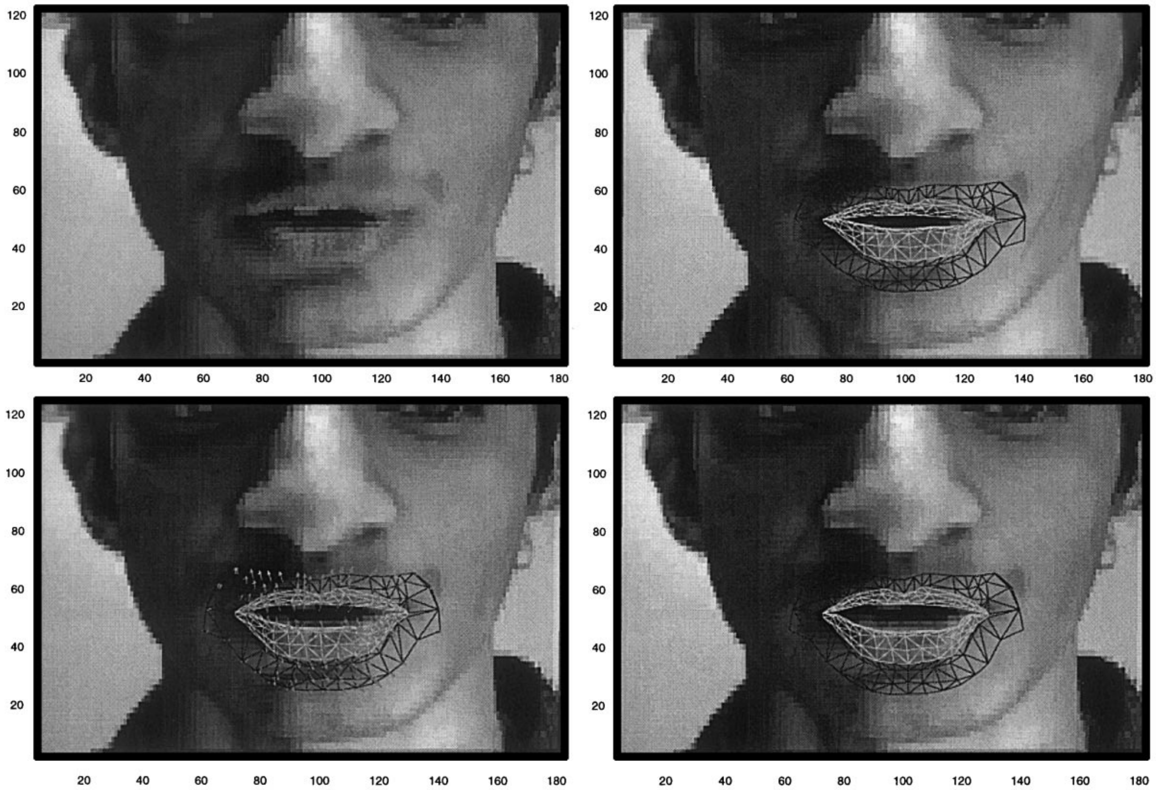

Fig. 8. From initial image to final fit.

available on the Elsevier website [2]. We recommend viewing these sequences both at normal speed, in order to see the natural quality of the motions, but also at reduced speed, in order to see the quality of the fit from frame to frame.

6.3. Advantages of the modal representation

It is worth noting here the flexibility that we gain from having a modal representation. As we have already described, the first few modes account for the greatest variation in lip shape, whereas the last few contribute the least. The more modes we use, the more accurately we can fit the shape (though the gain in accuracy drops with every mode). The fewer modes we use, the more robustly we can reject noise, since we only move along the directions of the greatest variation. Increasing or decreasing the number of modes we use for tracking is thus like moving a slider between accuracy (many modes) and robustness (few modes).

One of the main reasons we have used a small number of modes (10) thusfar is to be robust to noisy data. When clean data is available, though, we expect to be able to fit many more modes for a higher accuracy of fit. This points towards a much more convenient method of continuing to train the model when clean data is available. We can produce such clean data by carefully controlling the lighting or by improving colorspace separation by using colored lipstick. When such “clean” color data is available, we can fit many more than ten modes with high accuracy. Because point marking and tracking is then no longer necessary, we can easily train on large volumes of data in this way. We can then use data collected with this technique

[2] Speech files available. See http://www.elsevier.nl/locate/specom.

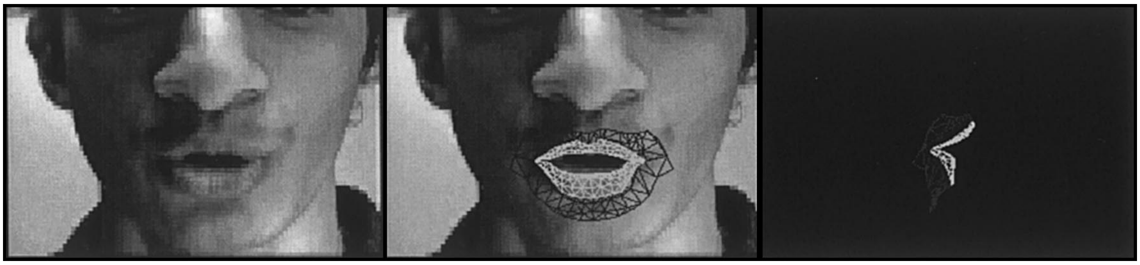

Fig. 9. Initial image, final fit and 3D reconstruction.

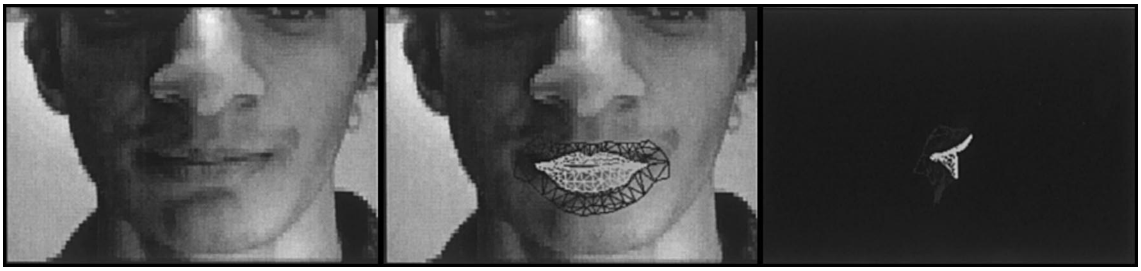

Fig. 10. Initial image, final fit and 3D reconstruction.

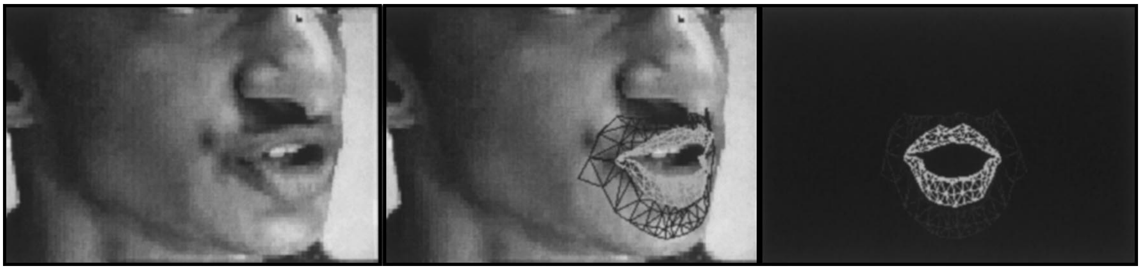

Fig. 11. Initial image, final fit and 3D reconstruction.

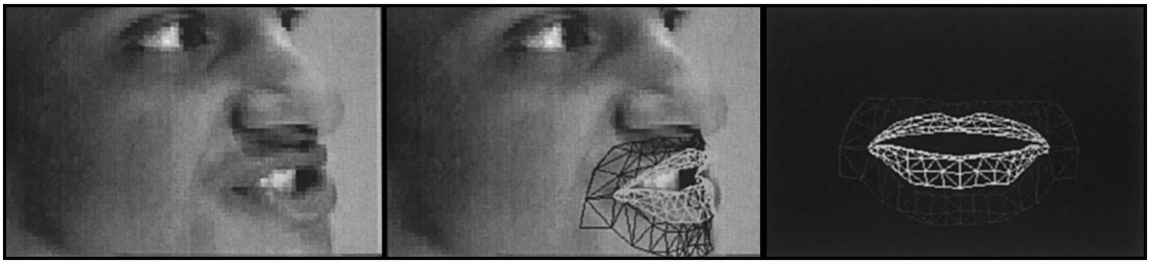

Fig. 12. Initial image, final fit and 3D reconstruction.

to find an improved set of generic modes (modeling the lip variations of many users) or a specialized set of modes (i.e., for high-detail tracking of an individual user).

On the other hand, in a video coding application, we may be receiving very noisy data (from a low quality camera) and we want to keep the number of transmitted bytes very low. For such an application, we want to use an even smaller number of modes (perhaps only two or three) to maximize robustness and data reduction.

The modal representation thus gives us the powerful capability of moving smoothly between high accuracy and high robustness, allowing us to adapt to the quality of the available data and the bandwidth of the communication channel.

7. Conclusions and future directions

We have presented a method for estimating and reconstructing the 3D shape of human lips from raw video data. This method began with a physical model with generic physical properties – a rubber sheet in the shape of the lips. 3D observations were then used to train this intial model with the true variations of human lip shapes, smoothly going from a physical to a hybrid physical–statistical model. This model fit naturally into a MAP estimation framework, which was then used for tracking and resynthesis of 3D lip shapes. We have shown through static and video examples how the observations in raw data can be accurately tracked, and have also demonstrated the ability of our model to accurately reconstruct 3D lip shapes from sparse 2D data. With the method presented here, we can now accurately and robustly track 3D lip shapes from 2D video data taken from an arbitrary pose.

This capability allows us to look towards a number of new directions in the future. The first step will be to integrate a 3D head pose tracker so that the model does not need to be initialized by hand in the first frame and so it can deal with large head motions within a sequence. The next step will be to incorporate this model as a feature of an audio-visual speech recognition system, allowing users to have truly unconstrained head motion while speaking. Other directions we wish to pursue include applying this method to other non-rigid areas of the face, such as the eyes and eyebrows, so that we can track and resynthesize the entire face *without* marking it with points. The methods described here can then be used for analysis/synthesis of all facial motion, which leads to many video coding and computer graphics applications.

Acknowledgements

This paper is dedicated to the memory of our friend and colleague Christian Benoît. His enthusiasm played a key role in our continued efforts in this research. In addition, the first author would like to thank the National Science Foundation for their support in the form of a Graduate Research Fellowship. The second author would like to thank La Caixa Foundation. Thanks also to Viewpoint Data Labs, who provided the model from which the initial model of the lips in Fig. 1 was extracted.

Appendix A. Using the finite element method

In this appendix, we give an overview of the finite element method (for the static equilibrium case), the specifics of our model, and the steps involved in applying our model to the lips.

A.1. The finite element method

The FEM is a numerical method for approximating the physics of an arbitrarily complex body. The central idea is to break the body into many small pieces whose individual physics are simple and then assembling the individual physics into a complete model. In contrast to the finite difference method, the finite element method also models the physics of the material *between* the nodes. This is possible because the method gives us interpolation functions that let us describe the entire modeled body in a piecewise analytic manner. Given this piecewise representation, the constitutive relations are *integrated* over the entire body to find the overall stress–strain relationship. These interpolation functions are written in vector form as

$$u(x) = \boldsymbol{H}(\boldsymbol{x})u, \tag{26}$$

where u represents the values of the function to be interpolated at the nodes, $\boldsymbol{H}(\boldsymbol{x})$ is the interpolation matrix, and $u(x)$ is the analytic representation of the function in the local coordinates of the element. In our case, the function we are interested in is the strain ϵ (internal force) resulting from a given deformation. We can find this using the relation

$$\epsilon(x) = \boldsymbol{B}(\boldsymbol{x})u, \tag{27}$$

where u now represents the displacements at each node and $\boldsymbol{B}$ is a combination of $\boldsymbol{H}(\boldsymbol{x})$ above and the stress–strain relationship of the material. It can be obtained by appropriately differentiating and recombining the rows of $\boldsymbol{H}$ given the stress modes specified by ϵ. To find the stiffness matrix $\boldsymbol{K}$ for an entire element, we integrate this relationship over the volume of the element,

$$\boldsymbol{k}_e = \int \boldsymbol{B}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{B}\,\mathrm{d}V, \tag{28}$$

where $\boldsymbol{C}$ describes the stress–strain relationship between the different degrees of freedom in the element. For each such matrix, we have the relationship

$$\boldsymbol{k}_e u = f, \tag{29}$$

where u represents the nodal displacements and f the strain resulting from these displacements. Note that the stresses and the strains are both expressed in the local coordinate system at this point. Each of these element matrices can be transformed by a matrix λ, which transforms the global coordinate system to the local one,

$$\lambda = \begin{bmatrix} \leftarrow & \hat{\imath} & \rightarrow \\ \leftarrow & \hat{\jmath} & \rightarrow \\ \leftarrow & \hat{k} & \rightarrow \end{bmatrix}, \tag{30}$$

where $\hat{\imath}$, $\hat{\jmath}$ and $\hat{k}$ are unit vectors in the local x, y and z directions, respectively. Because these vectors are orthonormal, λ^{-1} is simply the transpose of the above matrix. λ^{T} thus transforms from the local coordinate system to the global one.

Note that the matrix above transforms only three degrees of freedom: to apply it to the strain matrix for an entire element (which is nine-by-nine), we must repeat the same λ in the following block-diagonal form:

$$\boldsymbol{T} = \begin{bmatrix} \lambda & & \\ & \lambda & \\ & & \lambda \end{bmatrix}. \tag{31}$$

The matrix $\boldsymbol{T}$ can then be applied to the element strain matrix to produce the strain matrix in the global coordinate system

$$\boldsymbol{k}_{\mathrm{e}}' = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{k}_e\boldsymbol{T}. \tag{32}$$

In the expanded form on the right-hand side, we can see how in a vector post-multiplication (by a global displacement) this $\boldsymbol{k}_e'$ first transforms the vector to the local coordinate system (with $\boldsymbol{T}$), applies the stress–strain relation (with $\boldsymbol{k}_e$), and transforms the resulting force back into the global coordinate system (with $\boldsymbol{T}^T$).

The resulting transformed strain matrices now have aligned degrees of freedom and can be assembled into a single, overall matrix such that

$$\boldsymbol{K}u = f, \tag{33}$$

where the displacements and forces are now in the global coordinate system.

Further details of this method are described in many references on finite elements including (Bathe, 1982; Zienkiewicz and Cheung, 1967). Note that at the current time, we are not considering the higher order effects of dynamics (the mass and damping matrices) and thus do not describe them here.

A.2. Model specifics

For this application, a thin-shell model was chosen. We constructed the model by beginning with a 2D plane-stress isotropic material formulation (Zienkiewicz and Cheung, 1967) and adding a strain relationship for the out-of-plane components. For each triangular element, then, the six inplane degrees of freedom are related with a six-by-six matrix $\boldsymbol{k}_{xy}$, while the out-of-plane degrees of freedom are related by the three-by-three $\boldsymbol{k}_z$. In order to preserve the linearity of our model while maintaining the use of flat elements, we treat these two modes as being decoupled. They are thus assembled into the total $\boldsymbol{k}_e$ as shown in block-matrix form below:

$$\boldsymbol{k}_e = \begin{bmatrix} \boldsymbol{k}_{xy} & \\ & \boldsymbol{k}_z \end{bmatrix}. \tag{34}$$

We built the 2D $\boldsymbol{k}_{xy}$ using the formulation as described by Zienkiewicz and Cheung (1967) and Bathe (1982). This formulation has the following stress modes:

$$\epsilon = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \partial u/\partial x \\ \partial v/\partial y \\ \partial u/\partial y + \partial v/\partial x \end{bmatrix}, \tag{35}$$

where u and v correspond to displacements along the x and y dimensions of the local coordinate system. Using the relation $\epsilon(x) = \boldsymbol{B}(\boldsymbol{x})u$ and the overall displacement vector

$$u_e = [u_1 \quad v_1 \quad u_2 \quad v_2 \quad u_3 \quad v_3]^{\mathrm{T}}, \tag{36}$$

we can solve for $\boldsymbol{B}$. In addition, the material matrix $\boldsymbol{C}$ is

$$\boldsymbol{C} = \frac{E(1-v)}{(1+v)(1-2v)} \begin{bmatrix} 1 & \frac{v}{1-v} & 0 \\ \frac{v}{1-v} & 1 & 0 \\ 0 & 0 & \frac{1-2v}{2(1-v)} \end{bmatrix}, \tag{37}$$

where E is the elastic modulus and v is Poisson's ratio. For the lip model, Poisson's ratio was chosen to be 0.01. Since the elastic modulus E is a constant multiplier of the entire material matrix, it can be used to vary the stiffness of the element as a whole. As a result, a default value of 1.0 was used for this parameter. Elements that were to be more or less stiff than the default material were then assigned larger and smaller values, respectively.

The next step is to relate the out-of-plane degrees of freedom. It is important at this stage to consider the desired behavior of the material. If it were important for nodes to be able to move independently out of the plane without causing strain in the adjoined nodes, the $\boldsymbol{k}_z$ of Eq. (34) should be diagonal. In this case, however, it is desired that "pulling" on a given node has the effect of "pulling" its neighbors along with it. As a result, we construct the following $\boldsymbol{k}_z$:

$$\boldsymbol{k}_z = \frac{E(1-v)}{(1+v)(1-2v)} \begin{bmatrix} 1 & -0.5 & -0.5 \\ -0.5 & 1 & -0.5 \\ -0.5 & -0.5 & 1 \end{bmatrix}. \tag{38}$$

Consider this matrix in terms of the basic relation $\boldsymbol{k}_e u_e = f_e$. A positive displacement (out of the plane) in only one of these degrees of freedom produces negative forces (into the plane) in the other two. This means that stretching one node out of the plane without moving the other two *would require* forces pushing the other two down. When equal displacements are applied to all three nodes, there is zero strain on the element, resulting in a rigid motion mode out of the plane. Though the out-of-plane degrees of freedom are decoupled from the in-plane components, this mechanism acts to relate the strain energy to the deformation of the element due to out-of-plane displacements. The greater the disparity in out-of-plane displacements (i.e., the greater the stretching of the element due to such motions), the greater the strain energy is produced by this $\boldsymbol{k}_z$.

Once the degrees of freedom are properly aligned as described above, the resulting material has the approximate physics of many small plane-strain elements hooked together. The in-plane forces of one element can pull on both the in-plane and out-of-plane components of its neighbors, and the vice versa.

Once the complete $\boldsymbol{K}$ matrix has been assembled, we have a linear approximation to the relationship between the displacement and the resulting strain. We can now invert this relationship to find the displacements produced by an applied force (external strain). However, the matrix cannot be inverted as is: it is necessarily singular, as there are several displacement vectors that produce no stress in the body (i.e., they exist in the nullspace of $\boldsymbol{K}$). These are the modes of rigid motion. Consider, for example, a uniform displacement applied to all of the nodes. This would clearly produce no strain on the body. As a result, a minimum set of nodes must be "grounded" (i.e., held fixed) to prevent these singularities. For a 3D body, two nodes (6 DOF) must be grounded. This amounts to removing the rows and columns corresponding to the degrees of freedom for these nodes. The remaining $\boldsymbol{K}_s$ has full rank and can be inverted to provide the desired strain–stress relation:

$$\boldsymbol{K}_s^{-1} f_{\mathrm{s}} = u_{\mathrm{s}}, \tag{39}$$

while $\boldsymbol{K}$ is easy to compute and is band diagonal (due to the limited interconnections between nodes), finding its inverse is an expensive calculation. We thus want to take this inverse only once at a point where it is appropriate to linearize the physics.

A.3. Applying the method to the lips

The method described above can be directly applied to the mesh in Fig. 1, resulting in a physical model in that shape made up of a uniform elastic material. However, in order to accentuate certain properties of the lips, some additional information was added to the model. First, in order to maintain smoothness in the inner contours of the lips, the faces along the inside ridges of the lips were made twice as stiff as the default material. In addition, to allow relatively free deformation of the lips while still maintaining the necessary rigid constraints, a thin strip of low-stiffness elements was added to the top of the lips stretching back into the oral cavity. The nodes at the far end of this strip were then fixed in 3D. Lastly, since the FEM linearizes the physics of a body around a given point, the initial $\boldsymbol{K}$ matrix (Fig. 1) was used to deform the model to a more natural state of the lips (see Fig. 2), as described in the main text. The $\boldsymbol{K}$ and $\boldsymbol{K}_s^{-1}$ matrices used for the training were formed at this point to allow a more effective range for the linearized physics.

References

Adjoudani, A., Benoît, C., 1995. On the integration of auditory and visual parameters in an HMM-based ASR. In: Stork, D., Hennecke, M. (Eds.), Speechreading by Man and Machine, Springer, Berlin, pp. 461–472.

Basu, S. 1997. A three-dimensional model of human lip motion. Master's Thesis, MIT Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge.

Basu, S., Essa, I., Pentland, A., 1996. Motion regularization for model-based head tracking. In: Proc. 13th Internat. Conf. on Pattern Recognition, Vienna, 25–29 August 1996. IEEE Computer Society Press, Los Alamitos, Vol. C, pp. 611–616.

Basu, S., Oliver, N., Pentland, A., 1998. 3D modeling and tracking of human lips. In: Proc. Internat. Conf. on Computer Vision, Bombay, 4–7 January 1998. IEEE Computer Society Press, Los Alamitos, pp. 337–343.

Bathe, K., 1982. Finite Element Procedures in Engineering Analysis, Prentice-Hall, Englewood Cliffs.

Bregler, C., Omohundro, S., 1995. Nonlinear image interpolation using manifold learning. In: Tesauro, G., Touretzky, D., Leen, T. (Eds.), Advances in Neural Information Processing Systems, Vol. 7. MIT Press, Cambridge, pp. 401–408.

Coianiz, T., Torresani, L., Caprile, B., 1995. 2D deformable models for visual speech analysis. In: Stork, D., Hennecke, M. (Eds.), Speechreading by Man and Machine. Springer, Berlin, pp. 391–398.

Essa, I., 1995. Analysis and interpretation and synthesis of facial expressions. Ph.D. Thesis, MIT Department of Media Arts and Sciences, Massachusetts Institute of Technology, Cambridge.

Gelb, A. (Ed.), 1974. Applied Optimal Estimation, MIT Press, Cambridge.

Jebara, T., Pentland, A., 1997. Parametrized structure from motion for 3D adaptive feedback tracking of faces, In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition '97, San Juan, 17–19 June 1997. IEEE Computer Society Press, Los Alamitos, pp. 144–150.

Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active contour models. Internat. J. Computer Vision 1 (4), 321–331.

Lee, Y., Terzopoulos, D., Waters, K., 1995. Realistic modeling for facial animation. In: Proc. SIGGRAPH'95, Orlando, 6–11 August 1995. Addison-Wesley, Reading, MA, pp. 55–62.

Luettin, J., Thacker, N., Beet, S., 1996. Visual speech recognition using active shape models and hidden Markov models. In: Proc. Internat. Conf. on Acoustics Speech and Signal Processing 1996, Atlanta. IEEE Press, New York, pp. 817–820.

Martin, J., Pentland, A., Sclaroff, S., Kikinis, R., 1998. Characterization of neuropathological shape deformations. IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2), 97–112.

Oliver, N., Pentland, A., Bérard, F., 1997. LAFTER: lips and face real time tracker. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition '97, San Juan, 17–19 June 1997. IEEE Computer Society Press, Los Alamitos, pp. 123–129.

Summerfield, Q., 1979. Use of visual information for phonetic perception. Phonetica 36, 314–331.

Waters, K., Frisbie, J., 1995. A coordinated muscle model for speech animation. In: Davis, W., Prusinkiewicz, P. (Eds.), Graphics Interface '95. Canadian Human–Computer Communications Society, Ontario, pp. 163–170.

Zienkiewicz, O.C., Cheung Y.K., 1967. The Finite Element Method in Structural and Continuum Mechanics. McGraw-Hill, London.