LEARNING VISUAL MODELS FOR LIP READING

CHRISTOPH BREGLER
Computer Science Division
U.C. Berkeley
Berkeley, CA 94720
bregler@cs.berkeley.edu

AND

STEPHEN M. OMOHUNDRO
NEC Research Institute, Inc.
4 Independence Way
Princeton, NJ 08540
om@research.nj.nec.com

1. Introduction

This chapter describes learning techniques that are the basis of a "visual speech recognition" or "lipreading" system[1] . Model-based vision systems currently have the best performance for many visual recognition tasks. For geometrically simple domains, models can sometimes be constructed by hand using CAD-like tools. Such models are difficult and expensive to construct, however, and are inadequate for more complex domains. To do model-based lipreading, we would like a parameterized model of the complex "space of lip configurations". Rather than building such a model by hand, our approach is to have the system itself build it using machine learning. The system is given a collection of training images which it uses to automatically construct the models that are later used in recognition.

There are several phases of processing involved in our system. Ultimately, the recognition of the time sequence of images is performed using Hidden Markov Model technology similar to that used in speech recognition. Unlike speech recognition, however, there are extra phases to find,

[1]This is an extended version of [5].

track, and extract the lips from a sequence of images of a speaker. We will describe how the learned models are used to facilitate these tasks.

Some versions of our system do recognition based only on the visual input, while others use both visual and acoustic information. When visual and acoustic information is combined, it is necessary to deal with the fact that the acoustic sampling rate is higher than the visual image rate. We will describe how the learned models are used to interpolate between frames of video.

There is a common abstract learning task that underlies these different tasks. We use the expression "nonlinear manifold learning" for the task of inducing a smooth nonlinear surface in a high-dimensional space from a set of points drawn from that surface. This task is important throughout vision because it is often the case that the parameters of visual models are related by smooth nonlinear constraints. Learning such constraints and manipulating them in a computationally tractable way is therefore central to building learning-based visual recognition systems.

The first section of this chapter describes our representation for manifolds and the algorithm for learning it from data. The next two sections present experiments on learning synthetic models where performance can be directly evaluated. We then describe how the learned manifolds may be used for interpolation. We describe the "lip manifold" our system learns for visual speech recognition. We show how we use this for improving the performance of a snake-based lip tracker and for the task of interpolating between lip images. We present performance results for a single speaker using just visual information. Finally, we describe more complex experiments with multiple speakers and combined visual and acoustic information.

2. Smooth nonlinear manifold representation and induction

2.1. MOTIVATION

Human lips are geometrically complex objects whose shape varies with several distinct degrees of freedom that are controlled by the facial musculature of a speaker. For recognition, we would like to extract these degrees of freedom by using a computational representation of certain aspects of lip shape. If we represent a single configuration of the lips as a point in a feature space, then the set of all lip configurations achievable by a speaker will describe a smooth surface in the feature space. In differential geometry, such smooth surfaces are called "manifolds". For example, as a speaker smoothly opens her lips, the corresponding point in the lip feature space will move along a smooth curve. Changing the orientation of the lips would move the configuration point along a different curve in the feature space. Allowing both the lips to open and the orientation to vary would give rise to

configurations that describe a two-dimensional surface in the feature space. The dimension of the "lip" surface is the same as the number of degrees of freedom of the lips including both intrinsic degrees of freedom due to the musculature and external degrees of freedom representing properties of the viewing conditions.

We would like to learn such manifolds from examples and to perform the computations on them required by recognition. We may abstract the problem as the "manifold learning problem": given a set of points drawn from a smooth manifold in a space, induce the dimension and structure of the manifold.

There are a variety of tasks which are important to perform on such learned surfaces. Perhaps the most important such task for recognition is the "nearest point" query. The system must return the point on the surface which is closest to a specified query point (Fig. 1a). This task arises in any recognition context where the entities to be recognized are smoothly parameterized (eg. recognition of objects which may be rotated, scaled, etc.) There would be one surface for each class representing its feature values as the various parameters are varied [17]. Under simple noise models, the best classification choice for recognition will be to choose the class of the surface whose closest point is nearest the query point. The choice of surface gives the class of the recognized entity and the closest point itself provides the best estimate for values of the parameters of that entity. We will see that this query also arises in several other situations in a recognition system. We would therefore like the surface representation to support it efficiently in order to perform fast indexing.

Other important classes of queries are "interpolation queries" and "prediction queries". For these, two or more points on a curve are specified and the goal is to interpolate between them or extrapolate beyond them. Knowledge of the constraint surface can dramatically improve performance over "knowledge-free" approaches like linear or spline interpolation. (Fig. 1b)

Another important set of queries are "completion queries". In these queries, the values of certain features are unknown and have to be determined by the remaining features. The explicit manifold representation restricts the range of the unknown features based on the remaining features. (Fig. 1c). This task generalizes the usual task of regression. If the variables of the feature space are split into an "input" set and an "output" set, then the graph of a mapping between these sets is a surface. Evaluation of the mapping is equivalent to specifying the "input" set and letting the surface determine the "output" set. Unlike traditional regression methods, however, if the surface is represented explicitly we can also specify the "output" variables and ask for constraints on the "input" variables. This

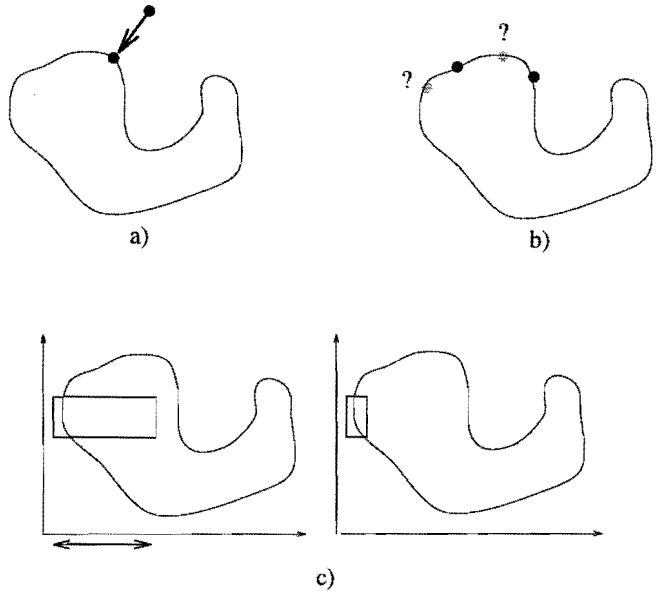

Figure 1. Surface tasks a) Closest point query, b) interpolation and prediction, c) generalized regression

allows inversion of learned mappings. Even more importantly, we can use this type of query to combine multiple constraints allowing each to further constrain the variables of the other.

2.2. MANIFOLD REPRESENTATION BASED ON THE CLOSEST POINT QUERY

In this chapter we describe a manifold representation based on the closest point query. Our approach starts from the observation that if the data points were drawn from a *linear* manifold, then we could represent it computationally by a point on the surface together with a projection matrix that projects arbitrary vectors orthogonally to the surface such that the resulting vector is parallel to the surface. Given a set of points drawn from such a surface, principal components analysis can be used to discover the dimension of the linear space and to find the best fitting projection matrix in the least squares sense. The largest principal vectors span the space and there is a precipitous drop in the principle values at the dimension of the surface (This is similar to approaches described [12, 22, 21]). A principal components analysis will no longer suffice, however, when the manifold is nonlinear because even a 1-dimensional curve could be embedded so as to span all the dimensions of the space.

If a nonlinear manifold is smooth, however, then each local piece ap-

pears more and more linear under magnification. If we consider only those data points which lie within a local region, then to a good approximation we may model them with a linear patch. The principal values can be used to determine the most likely dimension of the patch and that number of the largest principal components span its tangent space. The idea behind our representations is to "glue" such local patches together using a partition of unity (ie. a set of smooth non-negative functions which sum to one everywhere).

The manifold is represented as a mapping from the embedding space to itself which takes each point to the nearest point on the manifold. K-means clustering is used to determine an initial set of "prototype centers" from the data points. A principal components analysis is performed on a specified number of the nearest neighbors of each prototype point. These "local PCA" results are used to estimate the dimension of the manifold and to find the best linear projection in the neighborhood of prototype i. The influence of these local models is determined by Gaussians centered on the prototype location with a variance determined by the local sample density. The projection onto the manifold is determined by forming a partition of unity from these Gaussians and using it to form a convex linear combination of the local linear projections:

$$P(x) = \frac{\sum_i G_i(x) P_i(x)}{\sum_i G_i(x)} \tag{1}$$

This initial model is then refined to minimize the mean squared error between the training samples and the nearest surface point using EM optimization [6] and gradient descent.

A related mixture model approach applied to input-output mappings was presented by [9].

2.3. SYNTHETIC EXPERIMENTS

To test this approach, we generated sample sets from artificial manifolds and applied the learning technique to them. (Section 3 describes the application of the technique along with the closest point query to tracking and interpolation of real lip images.)

Figure 2a shows 200 sample points drawn from a one-dimensional curve in a two-dimensional space. 16 prototype centers are chosen by k-means clustering. At each center, a local principal components analysis is performed on the closest 20 training samples. Figure 2b shows the prototype centers and the two local principal components as straight lines. In this case, the larger principal value is several times larger than the smaller one. The system therefore attempts to construct a one-dimensional learned man-

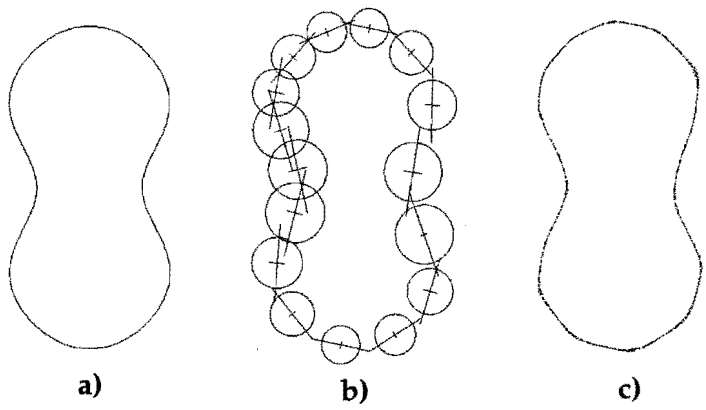

Figure 2. Learning a 1-dimensional surface. a) The surface to learn b) The local patches and the range of their influence functions, c) The learned surface.

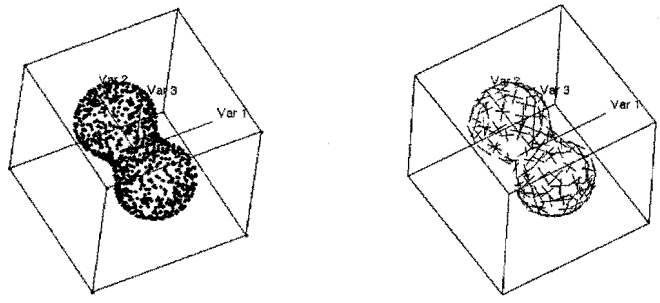

Figure 3. Learning a two-dimensional surface in the three dimensions a) 1000 random samples on the surface b) The two largest local principle components at each of 100 prototype centers based on 25 nearest neighbors.

ifold. The circles in Figure 2b show the extent of the Gaussian influence functions for each prototype. Figure 2c shows the resulting learned surface. It was generated by randomly selecting 2000 points in the neighborhood of the surface and projecting them according to the learned model.

Figure 3 shows the same process applied to learning a two-dimensional surface embedded in three dimensions.

2.4. CLOSEST POINT QUERY COMPARED TO NEAREST NEIGHBOR

To quantify the performance of this learning algorithm, we studied it on the task of learning a two-dimensional sphere in three dimensions. It is easy to compare the learned results with the correct ones in this case. Figure 4a shows how the empirical error in the nearest point query decreases

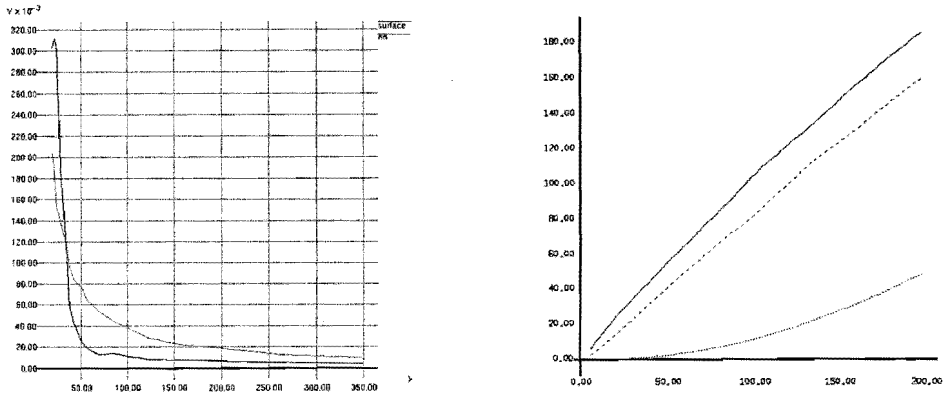

Figure 4. Quantitative performance on learning a two-dimensional sphere in three dimensions. a) Mean squared error of closest point querries as function of the number of samples for the learned surface vs. nearest training point b) The mean square root of the three principle values as a function of number of neighbors included in each local PCA.

as the number of training samples increases. It also shows the error made by a nearest-neighbor algorithm. With 50 training samples our approach produces an error which is one-fourth as large. Figure 4b shows how the average size of the local principal values depends on the number of nearest neighbors included. Because this is a two-dimensional surface, the two largest values are well-separated from the third largest. The rate of growth of the principal values is useful for determining the dimension of the surface in the presence of noise.

3. Using manifold representation for interpolation

So far we have described how to perform the closest point query on our learned manifold representation. We are also interested in interpolating between two given points using the manifold representation and the closest point query. Geometrically, linear interpolation between two points may be thought of as moving along the straight line joining the two points. This might cause the interpolated points to lie outside the space of reachable configurations. In our non-linear approach to interpolation, the point moves along a curve in the learned manifold that joins the two points. This constrains the interpolated point to only "meaningful" values. We have studied several algorithms for estimating the shortest manifold trajectory connecting two given points. For our performance results, we are interested in the point which is halfway along the shortest trajectory. We have studied three algorithms for finding a point on the surface which approximates this point.

3.1. "FREE-FALL"

The computationally simplest approach is to simply project the linearly interpolated point onto the nonlinear manifold. The projection is accurate when the point is close to the surface. In cases where the linearly interpolated point is far away (i.e. no weight of the partition of unity dominates all the other weights) the closest-point-query does not result in a good interpolant. For a worst case, consider a point in the middle of a circle or sphere. All local patches have same weight and the weighted sum of all projections is the center point itself, which is not a surface point. Furthermore, near such "singular" points, the final result is sensitive to small perturbations in the initial position.

3.2. "SURFACE-WALK"

A better approach is to "walk" along the surface itself rather than relying on the linear interpolant. Each step of the walk is initially taken to be linear and in the direction of the target point. The result of a step is immediately projected onto the manifold, however. The next step is then taken from this new point. When the target is finally reached, the arc length of the curve is approximated by the accumulated lengths of the individual steps. The point half way along the curve as measured by this arc length is then chosen. This algorithm is far more robust than the first one, because it only uses very local projections, even when the two input points are far away from each other. Figure 5b illustrates this algorithm.

3.3. "SURFACE-SNAKE"

In some ways this approach is a combination of the first two algorithms. It begins with linear interpolated points and iteratively moves the points toward the surface. The *Surface-Snake* is a sequence of n points preferentially distributed along a smooth curve with equal distances between them. An energy function is defined on such sequences of points so that the energy minimum tries to satisfy the three constraints of smoothness, equidistance, and nearness to the surface:

$$E = \sum_i \alpha ||v_{i-1} - 2v_i + v_{i+1}||^2 + \beta ||v_i - proj(v_i)||^2 \qquad (2)$$

E has value 0 if all v_i are equally distributed on a straight line and also lie on the surface. In general E can never be 0, if the surface is nonlinear, but a minimum for E represents an optimizing solution.

We begin with a straight line between the two input points and perform gradient descent in E to find this optimizing solution.

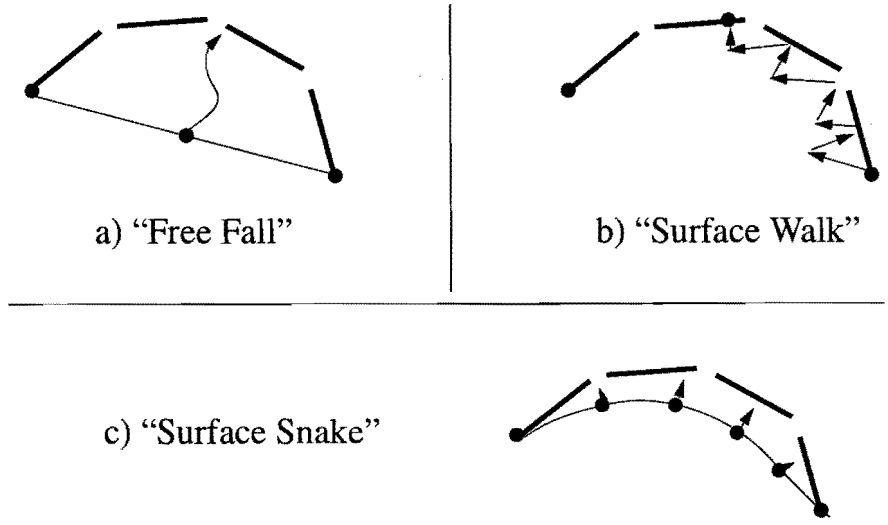

Figure 5. Proposed interpolation algorithms.

For another approach to nonlinear interpolation using a different architecture see [2].

3.4. SYNTHETIC EXPERIMENTS

To quantify the performance of these approaches to interpolation, we generated a database of $16 * 16$ pixel images consisting of rotated white bars on black background. The bars were rotated for each image by a specific angle. This represents a one-dimensional nonlinear surface embedded in a 256 dimensional image space. A nonlinear surface represented by 16 local linear patches was induced from the 256 images. Figure 6a shows two bars and their linear interpolation. Figure 6b shows the nonlinear interpolation using the *Surface-Walk* algorithm. The slider bars below the image represent the current weights for the linear patches which are necessary to produce the interpolated image.

Figure 7 shows the average pixel mean squared error of linear and nonlinear interpolated bars. The x-axis represents the relative angle between the two input points.

Figure 8 shows some iteration of a *Surface-Snake* interpolating 7 points along a 1 dimensional surfaces embedded in a 2 dimensional space.

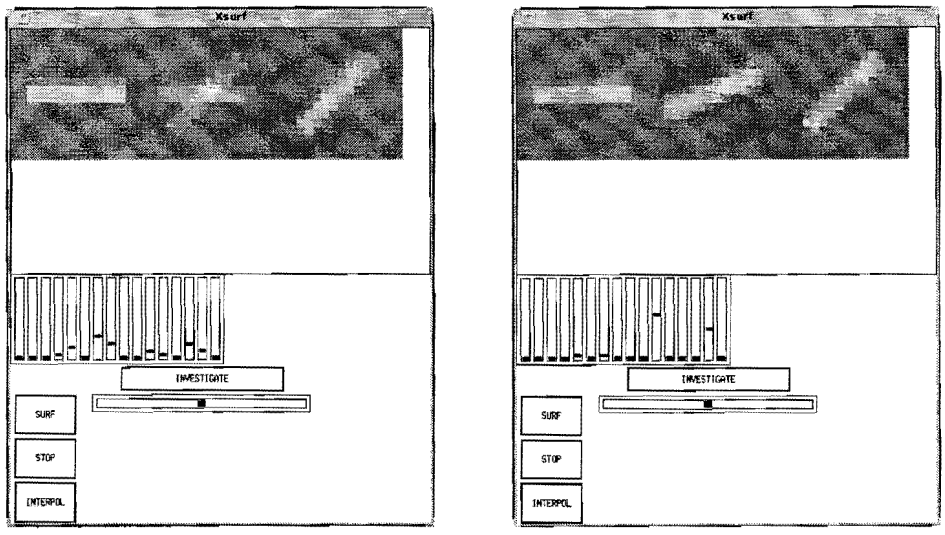

Figure 6. a) Linear interpolation, b) nonlinear interpolation.

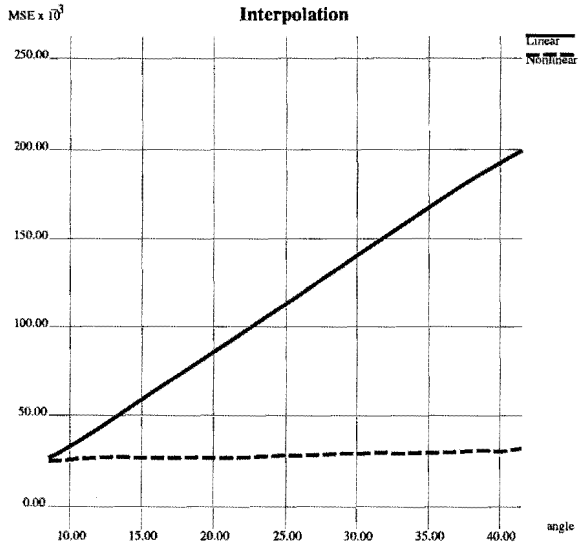

Figure 7. Average pixel mean squared error of linear and nonlinear interpolated bars.

4. Application to visual speech recognition

We are using the manifold techniques described above in a system for visual speech recognition. We view certain feature vectors of human lips as

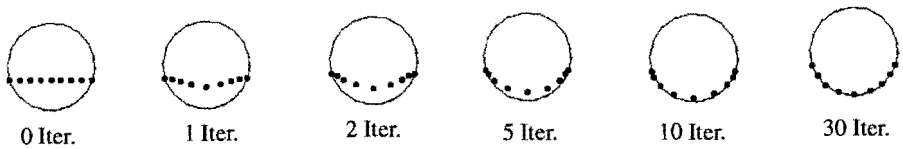

Figure 8. Surface-Snake iterations on an induced 1 dimensional surface embedded in 2 dimensions.

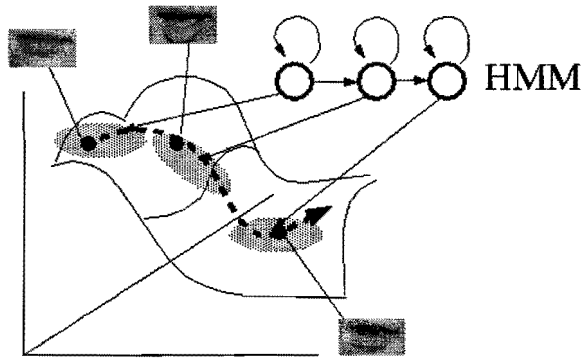

Figure 9. Lip trajectory approximated as HMM embedded in manifold representation

points which are constrained to lie on a low-dimensional nonlinear manifold embedded in the lip feature space. This manifold represents all possible lip configurations. While uttering a word or a sentence, the "lip feature point" moves along a trajectory on this manifold. (Fig. 9).

We model these trajectories using Hidden Markov Models (HMMs). An HMM represents a probability distribution over the space of strings over a specified set of output symbols. It consists of a set of hidden states and for each hidden state an emission distribution describing the probability of emitting each output symbol and a transition distribution describing the probability of making a transition to each hidden state. The hidden states and their transition distributions form an ordinary Markov model. An HMM generates a sequence of hidden states by starting in a start state and probabilistically transitioning to other states until an output state is reached. The corresponding string of symbols is generated by choosing a symbol according to the emission distribution of each hidden state in the sequence. The "forward-backward" algorithm is a dynamic programming algorithm for determining the probability of being in each hidden state at each location in an input string. It also produces the total probability that a given string was generated from a given HMM. The "Baum-Welch" algorithm is a technique based on EM for inducing the HMM parameters

from a set of sample strings.

Because of these properties, HMM's have been successfully applied to cryptography, speech recognition, protein modelling, and other applications. Because of the similarities with auditory speech recognition, we were motivated to try an HMM approach to visual speech recognition and to combined visual-acoustic speech recognition. The domain of the HMM emission vectors is defined by the lip-manifold. A specific HMM word model represents a probability distribution over trajectories on the lip-manifold for a given word. We represent the emission probability distributions by a mixture of gaussians or by a multi-layer-perceptron (MLP).

To get the input for the Hidden Markov Model we first find and track the lip position (section 4.1). We then extract the lip image at the selected location and size and code it as a point in a lip-feature space (section 4.2). When we want to perform combined acoustic and visual recognition, we fuse the visual n-dimensional visual feature vector together with a m-dimensional acoustic feature vector obtained from an acoustic frontend (section 4.4). Because the acoustic vectors are produced with a higher frame rate (necessary for good acoustic recognition), we need to interpolate the visual vectors (section 3). This produces a sequence of combined visual-acoustic $n + m$-dimensional vectors as input for the HMM.

The parameters of the HMM are set by the Baum-Welch procedure from a set of example utterances. We train a separate HMM for each word that is to be recognized. Once learned, the HMM's may be presented with a sequence of pure visual feature vectors or a sequence of bimodal visual-acoustic vectors. Each HMM estimates the likelihood that it generated the sequence and the most likely HMM is selected as the recognized utterance. In the pure visual domain we are interested in the recognition performance on the word level (section 4.3). In the visual-acoustic domain we are interested in the improvement that visual information can make to purely acoustic recognition in continuous speech recognition (section 4.5).

4.1. CONSTRAINT BOUNDARY TRACKING

To track the position of the lips we are using an "active vision" technique related to "snakes" [10] and "deformable templates" [25]. In each image, a contour shape is matched to the boundary of the lips. The space of contours that represent lips is represented by a learned lip-contour-manifold. During tracking we try to find the contour (manifold-point) which maximizes the graylevel gradients along the contour in the image.

The boundary shape is parameterized by the x and y coordinates of 40 evenly spaced points along the contour. The left corner of the lip boundary is anchored at $(0, 0)$ and all values are normalized to give a lip width

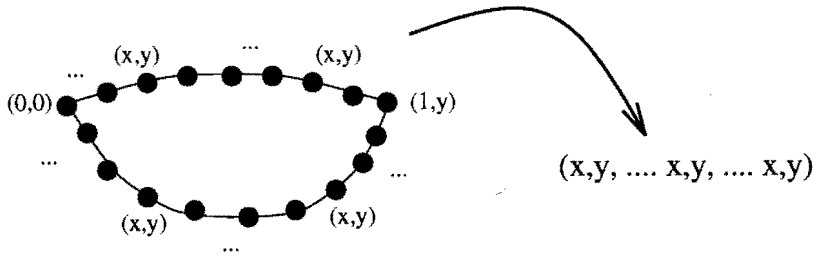

Figure 10. Lip contour coding

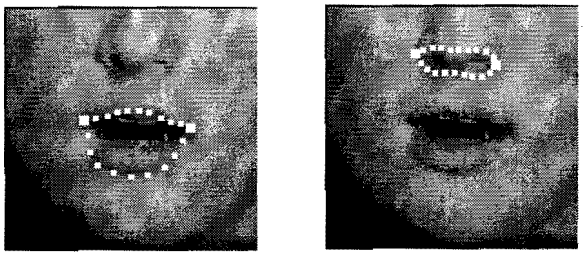

Figure 11. Snakes for finding the lip contours a) A correctly placed snake b) A snake which has gotten stuck in a local minimum of the simple energy function.

of 1 (Fig 10). Each lip contour is therefore a point in an 80-dimensional "contour-space" (because of the anchoring and scaling it is really a 77-dimensional space).

The training set consists of 4500 images of 6 speakers uttering random words. The training images are initially "labeled" with a conventional *snake* algorithm. The standard *snake* approach chooses a curve by trying to maximizing its smoothness while also adapting to certain image features along its length. These criteria are encoded in an energy function and the snake is selected by gradient descent. Unfortunately, this approach sometimes causes the selection of the boundary of incorrect neighboring objects (Fig. 11). To get a clean training sample, we cull the incorrectly aligned *snakes* from the database by hand.

We then apply the manifold learning technique described above to the database of correctly aligned lip snakes. The algorithm learns a 5-dimensional manifold embedded in the 80-dimensional contour space. 5 dimensions were sufficient to describe the contours with single pixel accuracy in the image. Figure 12 shows some of the lip models along two of the principal axes in the local neighborhood of one of the patches.

The tracking algorithm starts with a crude initial estimate of the lip position and size. In our training database all subjects positioned themselves

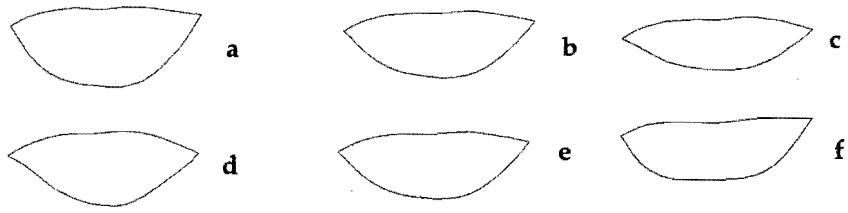

Figure 12. Two principle axes in a local patch in lip space. a, b, and c are configurations along the first principle axis, while d, e, and f are along the third axis.

at a similar location in front of the camera. The initial estimate is not crucial to our approach as we explain later. Currently work is in progress to integrate a full face finder, which will allow us to estimate the lip location and size without even roughly knowing the position of the subject.

Given the initial location and size estimate, we backproject an initial lip contour out of the lip-manifold back to the image (we choose the mean of one of the linear local patches). At each of the 40 points along the backprojected contour we estimate the magnitude of the graylevel gradient in the direction perpendicular to the contour. The sum of all 40 gradients would be maximal if the contour were perfectly aligned with the lip boundary. We iteratively maximize this term by performing a gradient ascent search over the 40 local coordinates. After each step, we anchor and normalize the new coordinates to the 80-dimensional shape space and project it back into the lip-manifold. This constraints the gradient ascent search to only to consider legal lip-shapes. Thus the search moves the lip-manifold point in the direction which maximally increases the sum of directed graylevel gradients. The initial guess only has to be roughly right because in the first few interations we use large enough image filters that the shape is attracted even far from the correct boundary.

The lip contour searches in successive images in the video sequence are initialized with the contour found from the previous image. Additionally we add a temporal term to the gradient ascent energy function which forces the temporal second derivatives of the contour coordinates to be small. Figure 13 shows an example gradient ascent for a starting image and the contours found in successive images.

4.2. LIP IMAGE CODING AND INTERPOLATION

In initial experiments we directly used the contour coding as the input to the Hidden Markov Models, but found that the outer boundary of the lips is not distinctive enough to give reasonable recognition performance. The inner lip-contour and the appearance of teeth and tongue are impor-

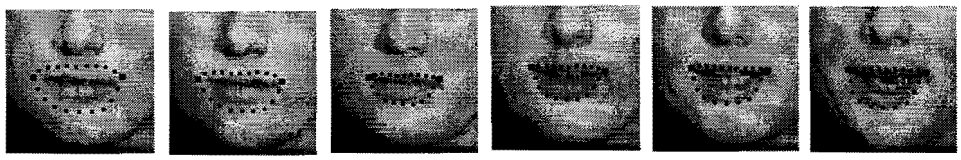

Figure 13. A typical relaxation and tracking sequence of our lip tracker

tant for recognition. These features are not very robust for tracking the lips, however, because they disappear frequently when the lips close. For this reason the recognition features we use consist of the components of a graylevel matrix positioned and sized at the location found by the contour based lip-tracker. Empirically we found that a matrix of 24x16 pixel is enough to distinguish all possible lip configurations. Each pixel of the 24x16 matrix is assigned the average graylevel of a corresponding small window in the image. The size of the window is determined by the size of the extracted contour. Because a 24x16 graylevel matrix corresponds to a 384-dimensional vector, we also reduce the dimension of the recognition feature space by projecting the vectors to a linear subspace determined by a principal components analysis.

To interpolate missing lip-images, we induce a nonlinear manifold embedded in this lower dimensional subspace. The interpolation is done in the lower dimensional linear space and is also constrained by the learned manifold. Figure 14 shows an example interpolation of lip images in a 32-dimensional linear subspace. Figure 14a shows the linear interpolation, and Figure 14b shows the nonlinear interpolation constrained by an 8-dimensional manifold, using the manifold-snake interpolation technique.

4.3. ONE SPEAKER, PURE VISUAL RECOGNITION

The simplest of our experiments is based on a small speaker dependent task, the "bartender" problem. The speaker may choose between 4 different cocktails names[2], but the bartender cannot hear due to background noise. The cocktail must be chosen purely by lipreading. A subject uttered each of the 4 words 23 times. An HMM was trained for each of the 4 words using a mixture of Gaussians to represent the emission probabilities. With a test set of 22 utterances, the system got only a single word wrong (4.5% error).

This task is artificially simple, because the vocabulary is very small, the system is speaker dependent, and it does not deal with continuous or spontaneous speech. These are all current state-of-the-art problems in the

[2]We use the words: "anchorsteam", "bacardi", "coffee", and "tequilla". Each word takes about 1 second to utter on average.

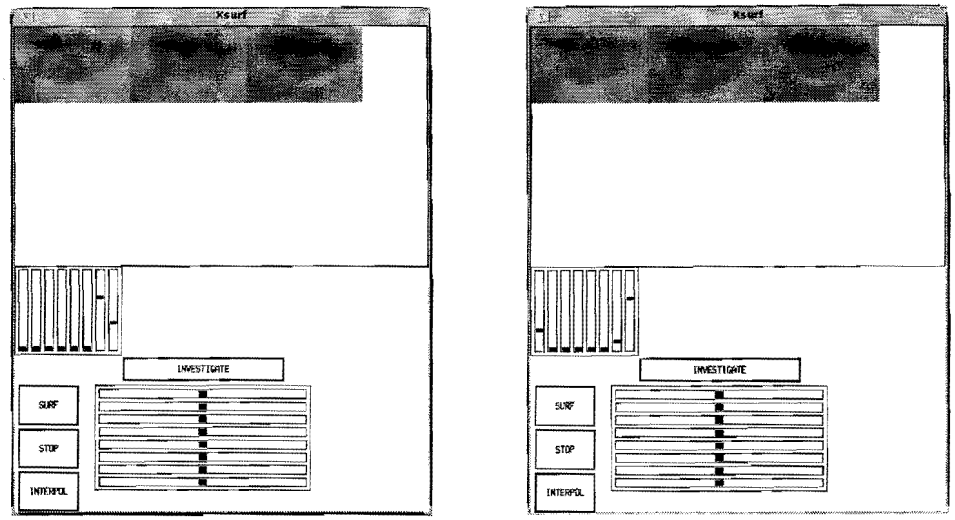

Figure 14. 24x24 images projected into a 32 dimensional subspace: a) linear interpolation b) nonlinear interpolation.

speech recognition community. For pure lip reading, however, the performance of this system is sufficiently high to warrant reporting. Subsequently we deal with the more state-of-the-art tasks using a system based on combined acoustic and visual modalities.

4.4. ADDITIONAL ACOUSTIC PROCESSING AND SENSOR FUSION

For the additional acoustic preprocessing we use an off-the-shelf acoustic front-end system, called RASTA-PLP [8] which extracts feature vectors from the digitized acoustic data with a constant rate of 100 frames per second.

Psychological studies have shown that human subjects combine acoustic and visual information at a rather high level. This supports a preceptual model that posits conditional independence between the two speech modalities [15]. We believe, however, that such conditional independence cannot be applied to a speech recognition system that combines modalities on the phoneme/viseme level. Visual and acoustic speech vectors are conditionally independent given the vocal tract position, but not given the phoneme class. Our experiments have shown that combining modalities at the input level of the speech recognizer produces much higher performance than combining them on higher levels.

4.5. MULTI-SPEAKER VISUAL-ACOUSTIC RECOGNITION

In this experiment we would like the visual lipreading system to improve the performance of acoustic speech recognition. We focus on scenarios where the acoustic signal is distorted by background noise or crosstalk from another speaker. Current state-of-the-art speech recognition systems perform poorly in such environments. Given the additional visual lip-information, we would like to determine how much error reduction can be achieved using the visual lip-manifold techniques.

We collected a database of 6 speakers spelling names or saying random sequences of letters. Letters can be thought of as small words, which makes this task a connected word recognition problem. Each utterance was a sequence of 3-8 letter names. The spelling task is notoriously difficult, because the words (letter names) are very short and highly ambiguous. For example the letters "n" and "m" sound very similar, especially in acoustically distorted signals. Visually they are more distinguishable (it is often the case that visual and acoustic ambiguities are complementary presumably for good evolutionary reasons). In contrast "b" and "p" are visually similar but acoustically different (voiced plosive vs. unvoiced plosive). With acoustic crosstalk from another speaker, the recognition and segmentation (i.e. when does one letter end and another begin) have additional difficulties. Information about which speaker's lips the acoustic signal is correlated to should make the recognizer more robust against crosstalk signals from other speakers.

Our training set consists of 2955 connected letters (uttered by the 6 speakers). We used an additional cross-validation set of 364 letters to avoid overfitting. In this set of experiments the HMM emission probabilities where estimated by a multi-layer-perceptron (MLP) [3]. The same MLP/HMM architecture has achieved state-of-the-art recognition performance on standard acoustic databases like the ARPA resource management task.

We have trained 3 different versions of the system: one based purely on acoustic signals using 9-dimensional RASTA-PLP features, and two that combine visual and acoustic features. The first bimodal system (AV) is based on the acoustic features and 10 additional coordinates obtained from the visual lip-feature space as described in section 4.2. The second bimodal system (Delta-AV) uses the same features as the AV-system plus an additional 10 visual "Delta-features" which estimate temporal differences in the visual features. The intuition behind these features is that the primary information in lip reading lies in the temporal change.

We generated several test sets covering the 346 letters: one set with clean speech, two with 10db and 20db SNR additive noise (recorded inside a moving car), and one set with 15db SNR crosstalk from another speaker

Task	Acoustic	AV	Delta-AV	rel. err.red.
clean	11.0 %	10.1 %	11.3 %	-
20db SNR	33.5 %	28.9 %	26.0 %	22.4 %
10db SNR	56.1 %	51.7 %	48.0 %	14.4 %
15db SNR crosstalk	67.3 %	51.7 %	46.0 %	31.6 %

TABLE 1. Results in word error (wrong words plus insertion and deletion errors caused by wrong segmentation)

uttering letters as well.

Table 1 summarizes our simulation results. For clean speech we did not observe a significant improvement in recognition performance. For noise-degraded speech the improvement was significant at the 0.05 level. This was also true of the crosstalk experiment which showed the largest improvement.

4.6. RELATED COMPUTER LIPREADING APPROACHES

One of the earliest successful attempts to improve speech recognition by combining acoustic recognition and lipreading was done by Petajan in 1984 [19]. More recent experiments include [14, 24, 4, 23, 7, 20, 16, 18, 13, 1, 11]. All approaches attempt to show that computer lip reading is able to improve speech recognition, especially in noisy environments. The systems were applied to phoneme classification, isolated words, or to small continuous word recognition problems. Reported recognition improvements are difficult to interpret and compare, because they are highly dependent on the complexity of the selected task (speaker dependent/independent, vocabulary, phoneme/word/sentence recogntion), how advanced the underlying acoustic system is, and how simplified the visual task was made (eg. use of reflective lipmarkers, special lipstick, or special lighting conditions). We believe that our system based on learned manifold techniques and Hidden Markov Models is the most complete system applied to a complex speech recognition task to date but it is clear that many further improvements are possible.

5. Conclusion and Discussion

This chapter can only begin to describe the many applications of manifold learning in vision. We have also not described certain hierarchical geometric data structures that can dramatically improve the computational performance of these techniques. We have shown how we are using them in the

domain of lip reading and that they give significantly improved recognition performance. It would be difficult to build traditional computer vision models of human lips and so the fact that our system builds these models by learning is significant. Many lip reading research groups mark a subject's lips with special reflective tape, paint, or lipstick or wire the subject with strain guages. The techniques described in this chapter show that such artifices are unnecessary and that video images may be directly used for visual speech recognition.

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 on Speechreading by Man and Machine*, 1995.
2. D. Beymer, A. Shashua, and T. Poggio. Example based image analysis and synthesis. *M.I.T. A.I. Memo No. 1431*, Nov 1993.
3. H.A. Bourlard and Morgan N. *Connectionist Speech Recognition, A Hybrid Approach.* Kluwer Academic Publishers, 1993.
4. C. Bregler, H. Hild, S. Manke, and A. Waibel. Improving connected letter recognition by lipreading. In *Int. Conf. Acoustics, Speech, and Signal Processing*, volume 1, pages 557–560, Minneapolis, 1993. IEEE.
5. C. Bregler and S.M. Omohundro. Nonlinear manifold learning for visual speech recognition. In W. Eric L. Grimson, editor, *Proceedings of the Fifth International Conference on Computer Vision*, pages 494–499, 10662 Los Vaqueros Circle, P.O. Box 3014, Los Alamitos, CA 90720-1264, June 1995. IEEE Computer Society Press.
6. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39, 1977.
7. Alan J. Goldschen. *Continuous Automatic Speech Recognition by Lipreading.* PhD thesis, Dept. of Electrical Engineering and Computer Science, George Washington University, 1993.
8. H. Hermansky, N. Morgan, A. Bayya, and P. Kohn. Rasta-plp speech analysis technique. In *Proc. ICASSP*, San Francisco, 1992.
9. M.I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2), March 1994.
10. Michael Kass, Andrew Witkin, and Demetri Terzopoulus. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.
11. R. Kaucic, B. Dalton, and A. Blake. Real-time lip tracking for audio-visual speech recognition applications. In *4th European Conf. Computer Vision*, April 1996.
12. M. Kirby, F. Weisser, and A. Dangelmayr. A model problem in represention of digitial image sequences. *Pattern Recognition*, 26(1), 1993.
13. J. Luettin, N. A. Thacker, and S. W. Beet. Visual speech recognition using active shape models and hidden markov models. In *to appear in IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 1996.
14. Kenji Mase and Alex Pentland. Lip reading: Automatic visual recognition of spoken words. *Opt. Soc. Am. Topical Meeting on Machine Vision*, pages 1565–1570, June 1989.
15. Dominic W. Massaro and Michael M. Cohen. Evaluation and integration of visual and auditory information in speech perception. *Journal of Experimental Psychology: Human Perception and Performance*, 9:753–771, 1983.
16. Javier R. Movellan. Visual speech recognition with stochastic networks. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7. MIT press, Cambridge, 1995.

17. H. Murase and S.K. Nayar. Visual learning and recognition of 3-d objects from appearance. *Int. J. Computer Vision*, 14(1):5–24, January 1995.
18. L. Nan, S. Dettmer, and M. Shah. Visual lipreading using eigensequences. In *Proc. of the Int. Workshop on Automatic Face- and Gesture-Recognition, Zurich, 1995*, 1995.
19. Eric D. Petajan. *Automatic Lipreading to Enhance Speech Recognition.* PhD thesis, University of Illinois at Urbana-Champaign, 1984.
20. Peter L. Silsbee. Sensory integration in audiovisual automatic speech recognition. In *28th Annual Asilomar Conf. on Signals, Systems, and Computers*, pages 561–565, November 1994.
21. P. Simard, Y. LeCun, and J. Denker. Efficient pattern recognition using a new transformation distance. In *Advances in Neural Information Processing Systems*, 1993.
22. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
23. Greg J. Wolff, K. Venkatesh Prasad, David G. Stork, and Marcus E. Hennecke. Lipreading by neural networks: Visual preprocessing, learning and sensory integration. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 1027–1034. Morgan Kaufmann, 1994.
24. Ben P. Yuhas, Moise H. Goldstein, Terence J. Sejnowski, and Robert E. Jenkins. Neural network models of sensory integration for improved vowel recognition. *Proc. IEEE*, 78(10):1658–1668, October 1990.
25. Alan L. Yuille, David S. Cohen, and Peter W. Hallinan. Facial feature extraction by deformable templates. Technical Report 88-2, Harvard Robotics Laboratory, 1988.