15

Probabilistic Models of Verbal and Body Gestures

C. Bregler, S.M. Omohundro, M. Covell, M. Slaney,
S. Ahmad, D.A. Forsyth, J.A. Feldman

Abstract

This chapter describes several probabilistic techniques for representing, recognizing, and generating spatiotemporal configuration sequences. We first describe how such techniques can be used to visually track and recognize lip movements to augment a speech recognition system. We then demonstrate additional techniques that can be used to animate video footage of talking faces and synchronize it to different sentences of an audio track. Finally we outline alternative low-level representations that are needed to apply these techniques to articulated body gestures.

15.1 Introduction

Gestures can be described as characteristic configurations over time. While uttering a sentence, we express very fine grained verbal gestures as complex lip configurations over time, and while performing bodily actions, we generate articulated configuration sequences of jointed arm and leg segments. Such configurations lie in constrained subspaces and different gestures are embodied as different characteristic trajectories in these constrained subspaces.

We present a general technique called *Manifold Learning*, that is able to estimate such constrained subspaces from example data. This technique is applied to the domain of tracking, recognition, and interpola-

tion. Characteristic trajectories through such spaces are estimated using Hidden Markov Models. We show the utility of these techniques on the domain of visual acoustic recognition of continuous spelled letters.

We also show how visual acoustic lip and facial feature models can be used for the inverse task: facial animation. For this domain we developed a modified tracking technique and a different lip interpolation technique, as well as a more general decomposition of visual speech units based on *Visemes*. We apply these techniques to stock-footage of a Marilyn Monroe scene and a news cast, where our technique is able to automatically modify a given utterance to a new sentence.

The models of verbal gestures that we use in the lip recognition and facial animation domain use low-level appearance-based and geometric representations. Lips and faces produce a relative constrained set of such features, which can be learned from data. In contrast, articulated objects, like hand and full body configurations, produce a much more complex set of image and geometric features. Constrained subspaces and sequences could be learned at higher levels of abstractions. Lower level representations should be based on much weaker and general constraints. We describe extensions to our gesture recognition approach that employ such low-level probabilistic constraints to image sequences of articulated gestures, and we outline how these new techniques can be incorporated into high-level manifold and HMM based representations.

Section 15.2 describes the constrained manifold representation using a mixture model of linear patches and a maximum likelihood estimation technique. Section 15.3 demonstrates an application to constrained tracking, and Section 15.4 describes a system that learns visual acoustic speech models for recognizing continuous speech. In Section 15.5 we briefly outline how to use the constrained space to interpolate lip images and in Section 15.6 we introduce a new set of techniques on how to use visual acoustic models for the *inverse task* of facial animation. Section 15.7 outlines the alternative low-level representations and how we plan to apply this to probabilistic gesture models of human body actions.

15.2 Constrained Lip Configuration Space

Human lips are geometrically complex shapes which smoothly vary with the multiple degrees of freedom of the facial musculature of a speaker. For recognition, we would like to extract information about these degrees of freedom from images. We represent a single configuration of the lips as a point in a feature space. The set of all configurations that a speaker

may exhibit then defines a smooth surface in the feature space. In differential geometry, such smooth surfaces are called "manifolds".

For example, as a speaker opens her lips, the corresponding point in the lip feature space will move along a smooth curve. If the orientation of the lips is changed, then the configuration point moves along a different curve in the feature space. If both the degree of openness and the orientation vary, then a two-dimensional surface will be described in the feature space. The dimension of the "lip" surface is the same as the number of degrees of freedom of the lips. This includes both intrinsic degrees of freedom due to the musculature and external degrees of freedom which represent properties of the viewing conditions.

We would like to learn the lip manifold from examples and to perform the computations on it that are required for recognition. We abstract this 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 several operations we would like the surface representation to support. Perhaps the most important for recognition is the "nearest point" query: return the point on the surface which is closest to a specified query point (Fig. 15.1a). This task arises in any recognition context where the entities to be recognized are smoothly parameterized (e.g., objects which may be rotated, scaled, etc.) There is one surface for each class which represents the feature values as the various parameters are varied [233]. Under a distance-based noise model, the best classification choice for recognition will be to choose the class of the surface whose closest point is nearest the query point. The chosen surface determines the class of the recognized entity and the closest point gives the best estimate for values of the parameters within that class. The same query arises in several other contexts in our system. The surface representation should therefore support it efficiently.

Other important classes of queries are "interpolation queries" and "prediction queries". Two or more points on a curve are specified and the system must 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. 15.1b)

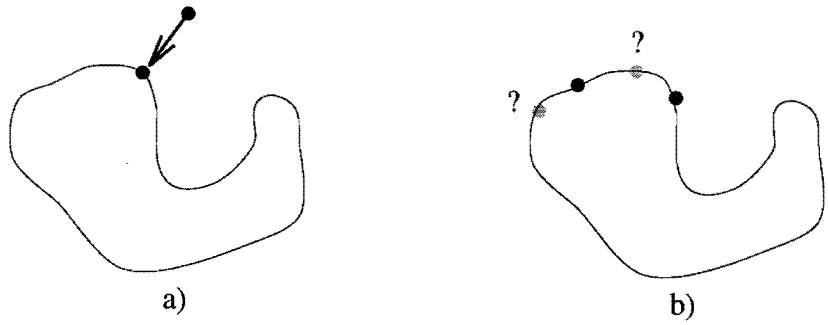


Fig. 15.1 Surface tasks a) closest point query, b) interpolation and prediction.

15.2.1 Mixtures of Local Patches

We present a manifold representation based on the closest point query [60]. If the data points were drawn from a *linear* manifold, then we could represent it by a point on the surface and a projection matrix. After the specified point is translated to the origin, the projection matrix would project any vector orthogonally into the linear subspace. Given a set of points drawn from such a linear surface, a principal components analysis could be used to discover its dimension and to find the least-squares best fit projection matrix. The largest principal vectors would span the space and there would be a precipitous drop in the principle values at the dimension of the surface (this is similar to approaches described [181, 301, 284]). A principal components analysis no longer suffices, however, when the manifold is nonlinear because even a one-dimensional nonlinear curve can span all the dimensions of a space.

If a nonlinear manifold is smooth, however, then each local piece looks more and more linear under magnification. Surface data points from a small local neighborhood will be well-approximated by a linear patch. Their principal values can be used to determine the most likely dimension of the patch. We take that number of the largest principal components to approximate the tangent space of the surface. The idea behind our representations is to "glue" such local linear patches together using a partition of unity.

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

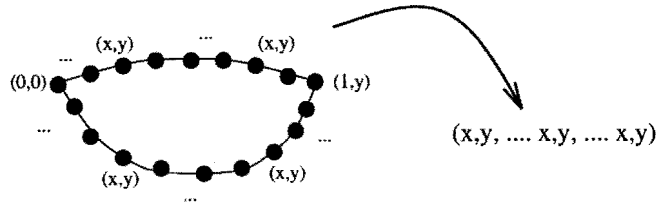


Fig. 15.2. Lip contour coding

"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:

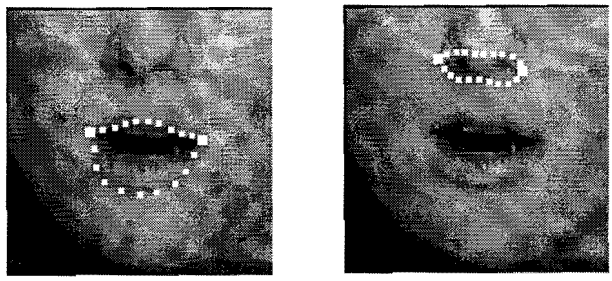
$$P(x) = \frac{\sum_i G_i(x) P_i(x)}{\sum_i G_i(x)} \tag{15.1}$$

This initial model is then refined to minimize the mean squared error between the training samples and the nearest surface point using EM optimization [105]. We have demonstrated the excellent performance of this approach on synthetic examples [59]. A related mixture model approach applied to input-output mappings appears in [163].

15.3 Constrained Tracking

To track the position of the lips we integrate the manifold representation with an "Active Contour" technique [172, 337, 195, 40]. 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 of 1 (Fig 15.2). Each lip contour is therefore a point in an 80-dimensional "contour-space" (because of anchoring and scaling it is actually only a 77-dimensional space).



a b

Fig. 15.3 Active contours 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.

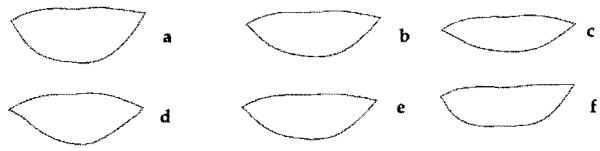


Fig. 15.4 Two principle axes in a local patch in lip space: a, b, and c are configurations along the first principle axis, while d, e, and f are along the third axis.

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 maximize its smoothness while also adapting to certain image features along its length. These criteria are encoded in an energy function and the snake is optimized by gradient descent. Unfortunately, this approach sometimes causes the selection of incorrect regions (Fig. 15.3). 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 15.4 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 them-

Fig. 15.5. A typical relaxation and tracking sequence of our lip tracker

selves at similar locations 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 knowing the rough position of the subject.

Given the initial location and size estimate, we backproject an initial lip contour from 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 constrains the gradient ascent search to only to consider legal lip-shapes. 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 the first few iterations use big enough image filters that the contour is attracted even far from the correct boundary.

The lip contour searches in successive images in the video sequence are started 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 15.5 shows an example gradient ascent for a starting image and the contours found in successive images.

15.4 Learning and Recognizing Temporal Lip Configuration Sequences with Hidden Markov Models

In initial experiments we directly used the contour coding as the input to the recognition 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 important for recognition. These features are not very robust for lip tracking, 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 pixels 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 dependent of the size of the found contour. Because a 24x16 graylevel matrix is equal 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.

15.4.1 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 cocktail names†, 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 made only one error (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 state-of-the-art problems in the speech recognition community. For pure lip reading, however, the performance of this system is sufficiently high to warrant reporting here. The following sections describe more state-of-the-art tasks using a system based on combined acoustic and visual modalities.

15.4.2 Acoustic Processing and Sensor Fusion

For the acoustic preprocessing we use an off-the-shelf acoustic front-end system, called RASTA-PLP [148] which extracts feature vectors from the digitized acoustic data with a constant rate of 100 frames per second.

† We choose the words: "anchorsteam", "bacardi", "coffee", and "tequilla". Each word takes about 1 second to utter on average.

Psychological studies have shown that human subjects combine acoustic and visual information at a rather high feature level. This supports a perceptual model that posits conditional independence between the two speech modalities [223]. 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.

15.4.3 Multi-Speaker Visual-Acoustic Recognition

In this experiment, the aim is to use the 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. State-of-the-art speech recognition systems perform poorly in such environments. We would like to know how much the additional visual lip-information can reduce the error of a purely acoustic system.

We collected a database of six 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 because of evolutionary pressures on language). In contrast, "b" and "p" are visually similar but acoustically different (voiced plosive vs. unvoiced plosive). Recognition and segmentation (when does one letter end and another begin) have additional difficulties in the presence of acoustical crosstalk from another speaker. Correlation with the visual image of one speaker's lips helps disambiguate the speakers.

Our training set consists of 2955 connected letters (uttered by the six speakers). We used an additional cross-validation set of 364 letters to avoid overfitting. In this set of experiments the HMM emission probabilities were estimated by a multi-layer-perceptron (MLP) [54]. The same MLP/HMM architecture has achieved state-of-the-art recognition

Task	Acoustic	AV	Delta-AV	relative 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 15.1 *Results in word error (wrong words plus insertion and deletion errors caused by wrong segmentation)*

performance on standard acoustic databases like the ARPA resource management task.

We have trained three different versions of the system: one based purely on acoustic signals using nine-dimensional RASTA-PLP features, and two that combine visual and acoustic features. The first bimodal system (AV) is based on the acoustic features and ten additional coordinates obtained from the visual lip-feature space as described in section 15.4. The second bimodal system (Delta-AV) uses the same features as the AV-system plus an additional ten 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 uttering letters as well.

Table 15.1 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.

15.4.4 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 [249]. More recent experiments include [217, 335, 57, 328, 138, 283, 230, 234, 208, 1, 175]. 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 (e.g., 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 one of the most complete systems applied to a complex speech recognition task to date but it is clear that many further improvements are possible.

15.5 Constrained Interpolation of Lip Sequences

So far we described how visual acoustic speech models can be used for recognition. In the next two sections we describe techniques that create new lip images which can be used for low-bandwidth video channels or facial animation applications.

First, we describe how the constrained manifold representation is applied to nonlinear image interpolation. This has applications to our domain of visual acoustic speech recognition where the different modalities are samples with different frequencies (30 images per second, 100 acoustic features per second). Another potential application of "model based" interpolation are video phone and video conference tasks, where the image frequency is usually lower then 30 frames per second.

Linear interpolated images are computed by traversing on a straight line between two key-feature vectors (images in our case). The interpolated image is the weighted average of two key images. Figure 15.6 shows an example image which is the average of an open mouth and a closed mouth. The knowledge about the space of "legal" mouth shapes should constrain interpolated images to only lie in this space, similar to our tracking task. We like to traverse along the shortest curve that is embedded in the nonlinear manifold. We experimented with different techniques on how to traverse between two points on a nonlinear manifold representation and achieved the best performance with a technique that we call "manifold snakes.

The technique begins with the linearly interpolated points and iteratively moves the points toward the manifold. The *Manifold-Snake* is a

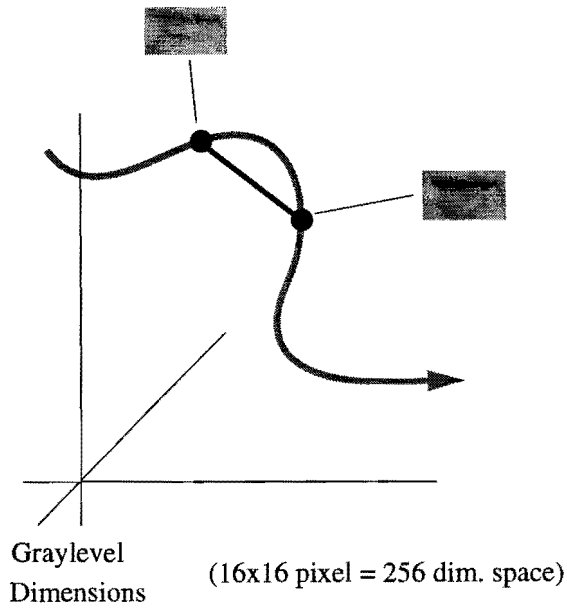


Fig. 15.6. Linear versus nonlinear interpolation.

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 these constraints (smoothness, equidistance, and nearness to the manifold):

$$E = \sum_i \alpha ||v_{i-1} - 2v_i + v_{i+1}||^2 + \beta ||v_i - proj(v_i)||^2 \qquad (15.2)$$

E has value 0 if all v_i are evenly distributed on a straight line and also lie on the manifold. In general E can never be 0 if the manifold 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 15.7 shows a case of linear interpolated and nonlinear interpolated 45×72 pixel lip images using this algorithm. The images were recorded with a high-speed, 100 frames per second camera†. Because of the much higher dimensionality of the images, we projected the images into a lower dimensional linear subspace. Embedded in this subspace we

† The images were recorded in the UCSD Perceptual Science Lab by Michael Cohen

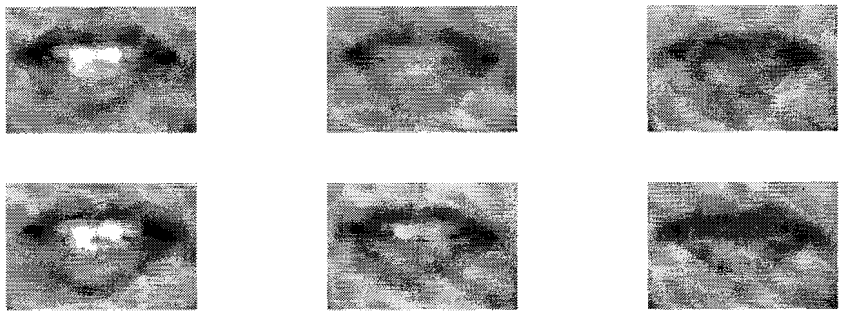


Fig. 15.7 45x72 images projected into a 16 dimensional subspace. Top row: linear interpolation. Bottom row: nonlinear "manifold-snake" interpolation.

induced a nonlinear manifold using a training set of 2560 images. The linearly interpolated lip image shows upper and lower teeth, but with smaller contrast, because it is the average image of the open mouth and closed mouth. The nonlinearly interpolated lip images show only the upper teeth and the lips half way closed, which is closer to the real lip configuration.

To learn the space of lip configurations in image space requires a relative large amount of example images. If we could code lip images using geometric features like we did for the tracking application, we could achieve a similar performance with less example shapes. Interpolating shapes is a "less nonlinear" task then interpolating graylevel images directly.

This leads to a different interpolation technique that is based on estimating geometric control points and using them for image morphing. We describe in the next section a facial animation system that uses such an alternative image interpolating technique.

15.6 Using Visual Acoustic Speech Models for Animation

The inverse problem to visual acoustic speech recognition is the speech animation problem. Traditionally, such systems are based on musculoskeletal models of the face that are driven by hand-coded dynamics [239]. Off the shelf text-to-speech systems produce phoneme categories that control the sequence of face model dynamics. Some systems are driven by input video data of tracked lips [292] or audio data [200] instead of handcoded heuristics, and some systems output modified video data [205, 276] instead of rendered graphics images.

We describe a system *VideoRewrite* that uses visual acoustic stock footage to build a video model of the face, and uses that model to repurpose video sequences of talking people so they can say new words. In the current version of the system we assume that the new acoustic utterance is given. The visual frames are generated such that they match the arbitrary new spoken sentence (visual dubbing). The system draws on the techniques that we introduced earlier for our visual-acoustic recognition and interpolation tasks. *VideoRewrite* can be described as an appearance based animation technique that is an alternative to traditional 3D face model and hand-coded dynamical model based graphics techniques.

15.6.1 Viseme Models

Our new experiments are applied to news cast and movie scenes with unconstrained vocabulary. This requires a more general decomposition of our visual acoustic speech models. If we wanted to recognize what has been said in the stock footage sequence, it would require modeling more than 60,000 words and we still would get a high error rate even with the best speech technology currently available. Instead to generate lip images synchronized to a new audio track we only need to model a small set of speech units that cover a basis set of lip movements. We developed a decomposition of 9 *viseme* categories that group together visual similar phonemes. This categorization is a modification of a *viseme* set introduced by [204]. Figure 15.8 shows example frames for each of the nine categories. For example the voiced plosive /b/ and the unvoiced plosive /p/ and the lip position of /m/ have a similar visual appearance and therefore are grouped together in one viseme category.

Besides a different speech decomposition the new domain also puts different requirements on our visual acoustic feature representation and estimation.

HMMs are generative models of speech. We could generate likely trajectories through the state space and emission probability distribution and then backproject manifold coefficients to image space using techniques that we developed for the constrained image interpolation task. Unfortunately, the range of lip configurations covered by a single HMM state is very large. This is alright for recognition but it is an obstacle for animation. In our experiments it produced very blurry images (the average of many possible poses and appearances for one viseme). Besides the constrained subspace representation, we also store

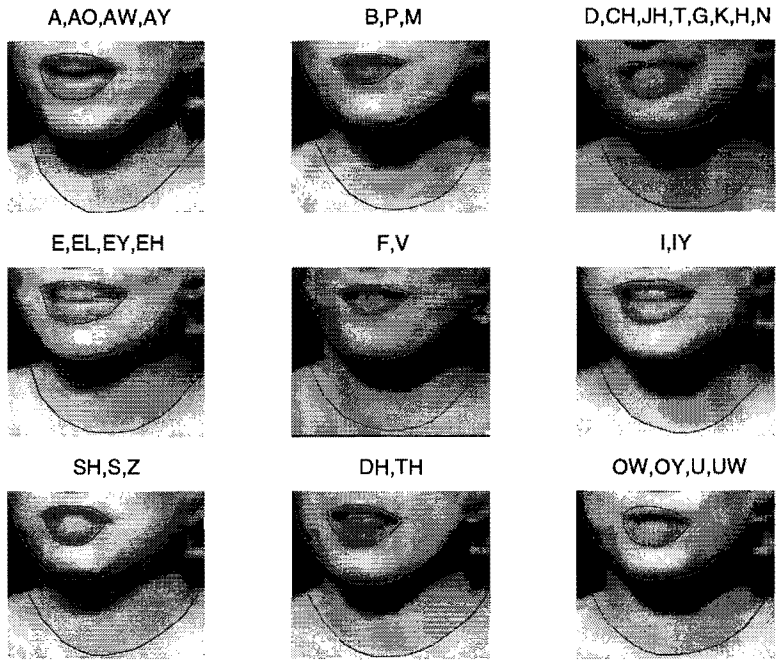


Fig. 15.8. Characteristic viseme example images.

the explicit lip-space coefficients or complete input images with tracked geometric control points for each viseme model. Interpolating these images produces sharper images then picking random (but likely) points in the HMM emission probability distribution. Another advantage of the explicit storage of a set of images and their control points is that in animation mode we have a choice between different pose modes. We describe in detail such techniques in the next subsection. Figure 15.9 illustrates the modified datastructure for our new viseme models. In some cases we model control points at the inner and outer lip contours and in some cases we also model points at the chin and neck, because they need to be animated for visual speech as well.

For acoustic features, we use the same front-end as in our earlier recognition experiments. The channel invariant coding of RASTA-PLP is useful because the stock footage and the new spoken utterance is most likely recorded using different microphone and room characteristics.

To model coarticulation we also experimented with bi-viseme models.

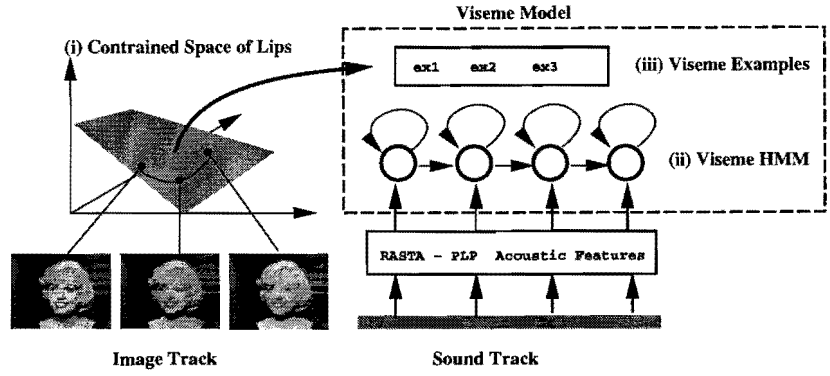


Fig. 15.9. Data structure of the Viseme Model

A bi-viseme is a pair of consecutive visemes. For example, saying a /ba/ or saying a /ma/ results in different /a/ lip positions.

Figure 15.9 illustrates our new visual acoustic speech models. Like the earlier models it consists of three main parts: (i) a constrained space for the lip configurations used for tracking, (ii) class based acoustic features modeled by the emission probability distribution of the HMM states, (iii) class based visual features modeled as a set of explicit image coefficients and control points.

15.6.2 Model Estimation

The estimation of the model parameters is done using two speech databases. To build the constrained lip/face tracking space and the viseme class based example set we use a stock-footage sequence of the desired person that we would like to animate. To estimate the acoustical parameters we decided to use a much larger speaker independent database that contains phonetic labels (TIMIT). The reason why we use two different datasets for the different modalities is to compensate for two problems: (a) some of the stock-footage sequences are too small to have enough training data for robust recognition, (b) usually we only have full-sentence "close captions" for the stock footage available, but no detailed phonetic labels that are necessary to train our viseme models.

The training of the constrained lip/face space using the stock-footage and the training of the acoustical parameters using the phonetic labeled TIMIT database is done using techniques described earlier. Based on the partially trained models we can estimate the viseme class-based ex-

ample lip configurations. The sentence based close-caption of the stock-footage is automatically transformed into a multiple pronunciation phonetic transcription using a pronunciation dictionary. The stock footage is decomposed into viseme sequences using the sound track and the trained acoustic HMMs in forced-viterbi mode. The lip and facial feature control points for each viseme image sequence are estimated using the trained lip/face space.

15.6.3 Appearance Based Animation

Now our visual acoustic speech model is ready for animation. It contains the trained acoustic features to transcribe the new input utterance, and a collection of example lip configurations for each viseme class to form an interpolated sequence of lip movements that fit the new audio track.

For the background, we need to pick an image sequence out of the stock footage that has at least the same length as the new utterance. The "background" stock-footage sequence is processed with the same constrained lip/face tracking algorithm as the training stock-footage to estimate the locations where we would like to change the facial parts.

We could just use a single background image that contains the rest of the non-moving facial parts, but we achieve a much more realistic video sequence if we retain the natural dynamics of the original scene. The sequences that we work with usually do not contain any drastic movements, but they never stay still. The head usually tilts to some extent and the eyes blink and produce various expressions. So far, we have made no attempt to synchronize these expressions to the new audio track. We replace just the lips or the lips, jaw, and neck to fit the new audio track. Potentially we could drive the other facial parts based on simple acoustic features. †

While the HMM models transcibe the new audio track, they index a corresponding sequence of example lip configurations. In case a set of alternative lip-examples is available, we can choose among different lip sequences. In that case, the one that best fits the estimated background sequence is choosen, using a metric for pose similarity and dynamic programming. Once we have the sequence of key-frames, we need to interpolate missing frames dependent on the rate of speech. We integrate the new lips into the original background sequence using the tracked contours of lips, chin, and neck. We call this "stitching". Figure 15.10

† For example the position of the eye-brows might change with pitch [237]. The system described in [84] models this finding.

shows the flow chart with example images. Figure 15.10(a) and (d) are two example key-frames. The images are spatially warped [25] in such a way that pixels along source contours are mapped to pixels along target contours. The two key frames have two different sets of source contours, but are mapped to the same set of target contours. To compute the target contour, we build the weighted average of the two key frame source contours and align the center and orientation of the upper lip contour with the original background lip contour. The lower lip contour and chin contour are rotated and shifted to the same extent, but not aligned to the contours of the background image. The neck contour is set equal to the background neck contour. Since the database includes the control-points the entire process is automatic.

Figure 15.10(b) and (c) shows the warped versions of the two key frames. The warped images are cross-faded and multiplied with a soft spatial mask, Figure 15.10(e)), before they are integrated into the background image.

A related technique based on optical flow measurements and image morphing was demonstrated for view interpolation of human faces by [28].

15.6.4 Experiments

We applied *VideoRewrite* to stock-footage of a 30 minute sequence of CNN Headline News, and a short sequence of a Marilyn Monroe movie. In the case of the CNN Headline News we had enough data to build the Full Bi-Viseme Database. The Marilyn Monroe database consists of the nine example visemes plus silence, shown in Figure 15.8. We dubbed both examples to TIMIT utterances and sentences recorded in our lab. Figure 15.11 shows an example of the original Marilyn Monroe sequence and the dubbed sequence. As you can see in some cases a closed mouth shape has to be modified to an open shape and vice versa. The position of the chin and the resulting shadow on the neck changes as well, because we don't align these contours. Figure 15.12 shows an example sequence of the CNN newscast anchor woman where we only modified the lips. In some cases, the chin moves to the different direction than the lips, which produces unrealistic motion.

The perceived realism of the animation is subjective. We believe it depends on the viewers lip-reading skills and the actor's/anchor's articulation skills. Highly trained lip-readers might have more objections to our dubbed video sequence than unexperienced viewers. Also Marilyn

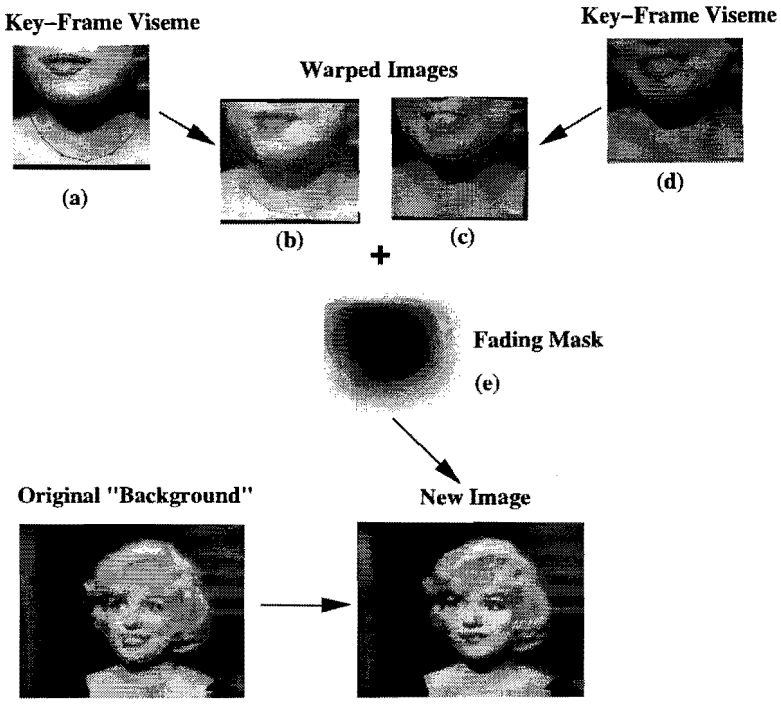


Fig. 15.10 Flow chart for morphing and stitching new lip images into the original movie sequence

Monroe tends to "overlap" most of her original utterances with a "large smile". Thus the range of articulation in Marilyn's speaking is limited.

Overall, we think the dubbed video sequences are more realistic than other animations produced by rendered 3D-models, text-to-speech systems, and hand coded articulators. An important feature of our system that increases the realism is the way we morph and blend the example database images into a background image sequence. Even if the rest of the face (eyes, eyebrows) moves in an uncorrelated way, the human observer usually gets the impression that such expressions fit the new utterance. So far we only modeled co-articulation with a very simple bi-viseme set. We believe a better treatment of co-articulation [84] would add another degree of realism.

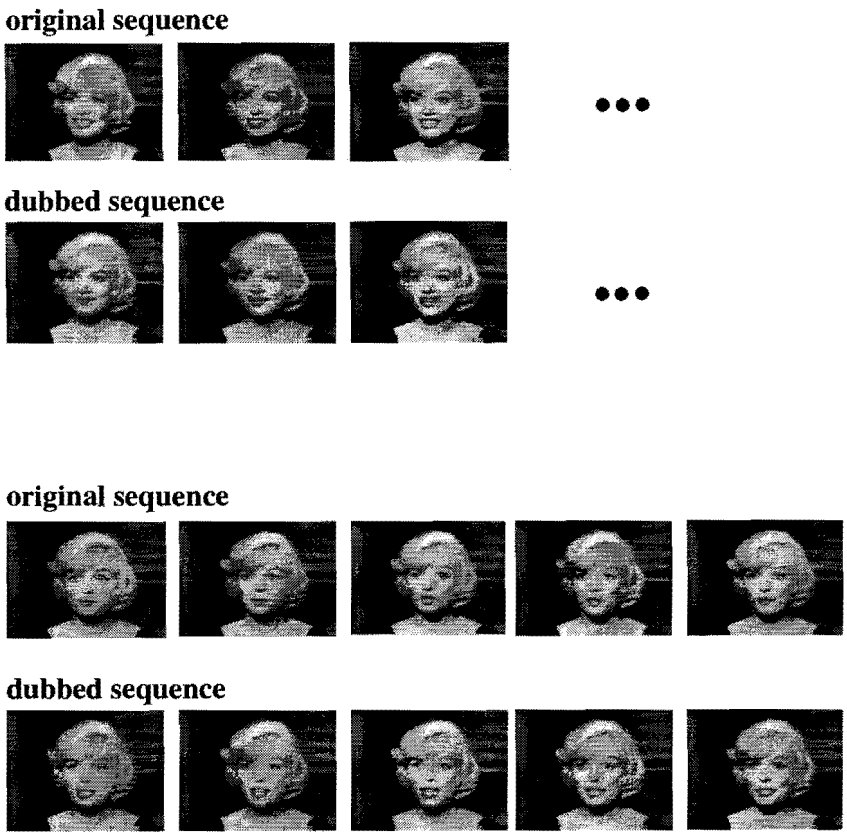


Fig. 15.11 Original and dubbed example images of the Marilyn Monroe scene using 10 visemes to model the lips, chin, and neck. The first sequence shows an open mouth being replaced with a closed-mouth, while the second sequence shows the opposite.

15.7 The Next Step: Recognizing Body Gestures

We described methods to represent and learn low-level feature constraints and temporal models of feature configurations. We applied these techniques to the domain of lip and facial features. The representations of geometric and appearances that we used are related to many other techniques applied to the same domain [301, 40, 341, 195]. Although faces and lips span a very complex set of configurations, the features generated lie on a relatively small constrained subspace.

This is different in the domain of articulated objects like human bodies or hands. Clothes generate a large range of appearance features

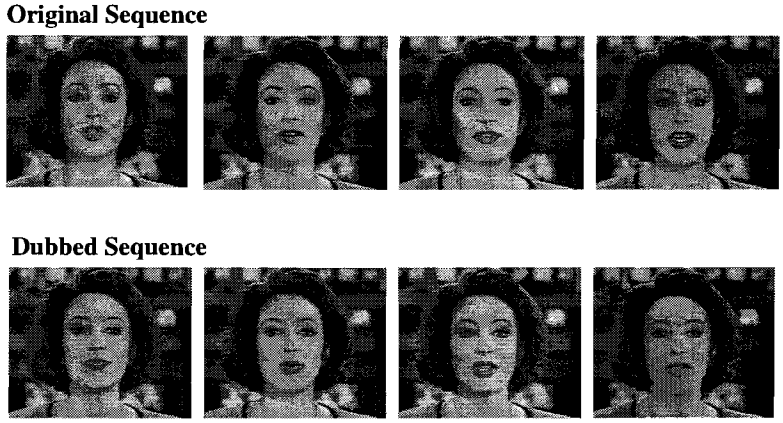


Fig. 15.12 Original and dubbed example images of CNN Headline News using 90 bi-visemes to model just the lips.

due to difference in color and texture. The large numbers of degrees of freedom and self-occlusion produces a large range of geometric based features. We believe that manifold based representations and Hidden Markov Models can be applied to learn constraints, higher level representations of articulated objects, like joint-angles, or body configurations over time. Lower level representations should be based on much weaker constraints, or more general properties.

We describe extensions to our gesture recognition approach that employ such low-level probabilistic constraints to image sequences of articulated gestures, and we outline how these new techniques can be incorporated into high-level manifold and HMM based representations.

The human body can be approximated by an assembly of rigid segments that are linked together at joints. While performing an action or gesture most of the body segments move most of the time. This is a single strong cue. The image region corresponding to a body segment contains a single coherent motion field. Two segments can be disambiguated by detecting two different coherent motion areas in the image. Joints can be detected if the pose and motion fields of a segment pair comply with the constraints associating with a body joint. Over multiple frames, characteristic sequences of jointed motion can be detected. For example, the process of walking consists of four connected body segments that traverse with three very characteristic periodic joint angle curves over time.

We introduce two low-level "layers" that represent single coherent mo-

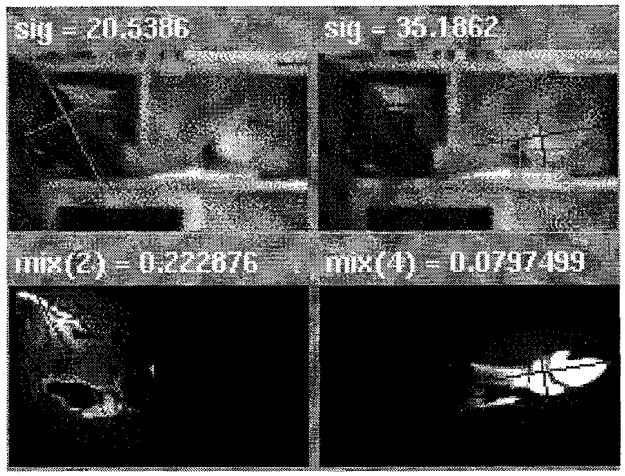


Fig. 15.13 Motion Coherence Blobs: Given two consecutive images we are able to group pixel regions with coherent motion and spatial proximity using Expectation Maximimzation (EM) search. blob models that model the lower and uper arm segments. The

tion blobs and jointed body segment pairs, and we describe how these representations are integrated into higher-level *kinematic manifolds* and HMM based dynamical models in a probabilistic framework. The problem of constrained estimation of body segments and their actions is described as a maximum a-posteriori estimation. The low-level hypothesis of coherent motion areas are the likelihood terms, and the higher-level kinematic and dynamical constaints are coded as priors.

Motion Coherence Likelihoods: A collection of body segments are described with a multidimensional mixture of Gaussian *Blobs*. For each blob model, the means describe the center of mass in the image and its affine motion parameters. Part of the covariance describes the spatial distribution, and one variance describes the graylevel deviation of the motion prediction given the previous image frame. Without the spatial parameters, this approach is similar to layered motion estimation using EM search [161, 12].

Each pixel has a *hidden random variable* that assigns this pixel to one of the motion blob models or a background model. We initialize these models with optical flow clustering and then perform a few EM steps. Figure 15.13 shows some examples.

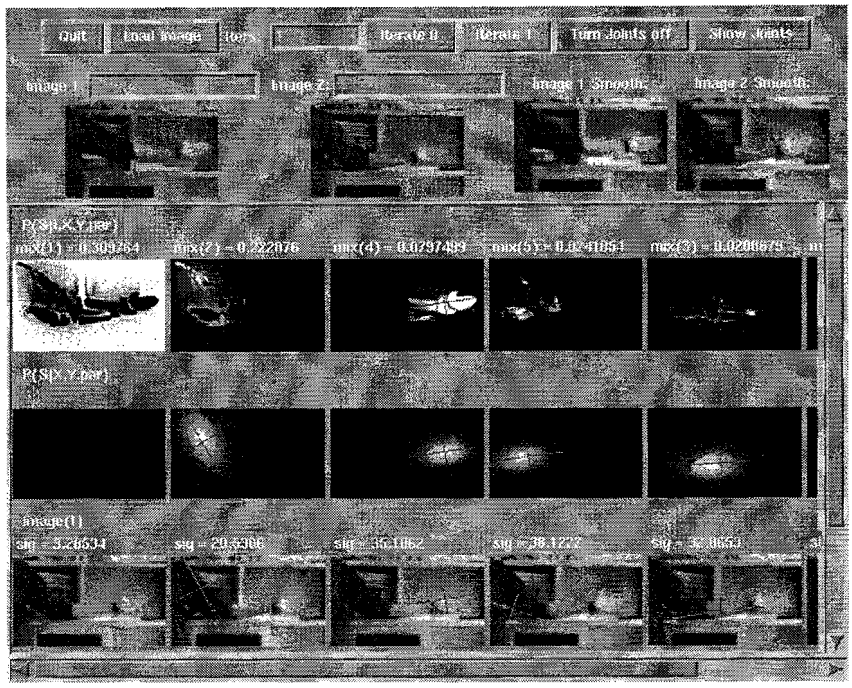


Fig. 15.14 One background model and four body segment hypotheses with joint constraints ranked with decreasing score values after two EM iterations. As you see the last 2 blob hypotheses have significant lower score that the first 2 hypotheses, because no joint hypotheses with compatible motion could be computed.

A similar representation based on *Blob* models for coherent color regions is applied to the human body domain by [329].

Simple Kinematic Priors: To further constrain the blob estimation and to incorporate high-level domain knowledge, we introduce body joint hypotheses and *score* coefficients for body segments and joints. General constraints can be coded straightforwardly as quadratic log-prior terms that give high values to body joint hypotheses at locations that are "compatible" with body segment pose and motion. The score parameter for each segment or joint is proportional to the fit of these constraints. For a more detailed description of the constraints see [56]. Figure 15.14 shows the top four ranked hypothesis of body segments and joints of an arm sequence.

Complex Kinematic Priors: A set of connected body segments with more than one joint, like a pair of legs, or the torso and arms complies to further constraints. Certain joint-angle configurations are not possible or occur less frequently than others. We can model and estimate such configuration scenarios with the manifold learning techniques described previously. A mixture of linear patches in the joint-angle space provides additional kinematic priors. Additional hidden random variables must be introduced to assign each of the body joint hypotheses to one of the linear patches. Another level of EM can be used for a feasible estimation process.

Dynamical Priors: Constraints over multiple frames that are *action specific* can be modeled with Hidden Markov Models. As in the manifold representation, additional hidden random variables assign each joint model to a Hidden Markov State. A bottom-up process estimates the expected value for this state, and then a top-down process uses priors from the Gaussian emission probability of the hidden states to further constrain the low-level estimation of body segment and joint models.

Figure 15.15 shows a typical set of blob hypotheses. The top row shows the computed curve of rotation differences between two jointed body segments. Hidden Markov Models that are trained for typical angle curves detect primitive actions like arm swings. More extensive experiments using these high-level priors are currently in progress.

15.8 Conclusion

We have shown how constrained configuration subspaces and temporal models for configuration sequences can be estimated from example data and used for recognition and animation. We applied such models to the domain of visual acoustic speech recognition and synthesis. We also outlined what additional low level feature constraints and mid-level articulated constraints are needed to estimate representations of articulated objects.

We believe that probabilistic modelling and learning such models from data is a crucial feature in our systems. Especially in the more complex domain of articulated human actions, we believe that bottom-up and top-down information flow, using iterative techniques like multiple EM optimizations, is another useful technique that shows how low and high-level models can interact and used to recognize non-trivial gestures.

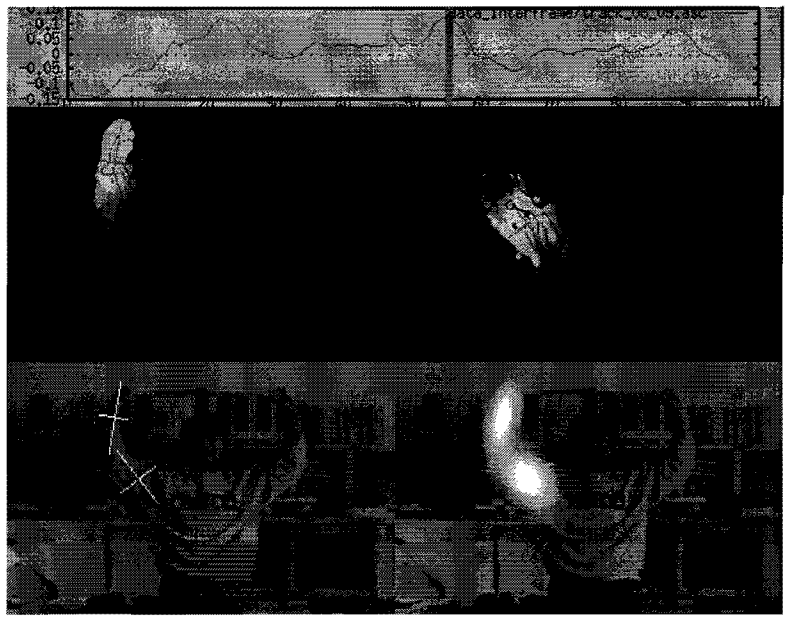


Fig. 15.15 The bottom row shows two motion blob models found for the left arm of the person. The middle-row shows the posteriori probabilities that a pixel belongs to either the upper arm blob or the lower arm blob. The top row shows a history of rotation angles. Each angle is the difference between the rotation of the upper and lower arm blob.

Acknowledgements

M. Bichsel
The author would like to thank Prof. P. Stucki for providing a stimulating working environment. He would also like to thank A.P. Pentland and Ch. Maggioni for a number of stimulating discussions.

B. Bascle, A. Blake and J. Morris
The authors gratefully acknowledge the financial support of EPSRC and the Oxford Metrics company.

C. Bregler, S.M. Omohundro, M. Covell, M. Slaney, S. Ahmad, D.A. Forsyth, J.A. Feldman
The authors would like to thank Yochai Konig, Jitendra Malik, Nelson Morgan, Jianbo Shi, Alex Waibel and many others in the Berkeley Speech and Vision Group, the CMU and Univ. Karlsruhe Interact group, and Interval Research for support and helpful discussions.

R. Cipolla and N.J. Hollinghurst
The authors gratefully acknowledge the financial support of EPSRC, and the donation of a robot manipulator by the Olivetti Research Laboratory, Cambridge and the collaboration of Mr. Masaaki Fukumoto and Dr. Yasuhito Suenaga of the NTT Human Interface Laboratories.

J. Cassell
Some of this work is carried out, and all of it is informed by my interactions with Mark Steedman, Norm Badler, Catherine Pelachaud, and the students of the Gesture-Jack group at the University of Pennsylvania, and Scott Prevost, Kris Thórisson, Hannes Vilhjalmsson,

Computer Vision for Human–Machine Interaction

Recent advances in the field of computer vision are leading to novel and radical changes in the way we interact with computers. It will soon be possible to enable a computer linked to a video camera to detect the presence of users, track faces, arms and hands in real time, and analyse expressions and gestures. The implications for interface design are immense and are expected to have major repercussions for all areas where computers are used, from the work place to recreation.

This book collects the ideas and algorithms from the world's leading scientists, offering a glimpse of the radical changes that are round the corner and which will change the way we will interact with computers in the near future.

Cover design: Struktur

Computer Vision for Human–Machine Interaction

CAMBRIDGE
UNIVERSITY PRESS
ISBN 0-521-62253-0
9 780521 622530

CAMBRIDGE

Computer Vision for

Edited by
Roberto Cipolla and A

COMPUTER VISION FOR HUMAN–MACHINE INTERACTION

Edited by
Roberto Cipolla and Alex Pentland

CAMBRIDGE
UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge CB2 1RP, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK http://www.cup.cam.ac.uk
40 West 20th Street, New York, NY 10011-4211, USA http://www.cup.org
10 Stamford Road, Oakleigh, Melbourne 3166, Australia

First published 1998

Printed in the United Kingdom at the University Press, Cambridge

Typeset by the author

Typeset in Computer Modern 10/13pt, in LaTeX

A catalogue record for this book is available from the British Library

ISBN 0 521 62253 0 hardback

Contents

Foreword: Out of Sight, Out of Mind

N. Negroponte

Face it. Butlers cannot be blind. Secretaries cannot be deaf. But somehow we take it for granted that computers can be both.

Human-computer interface dogma was first dominated by direct manipulation and then delegation. The tacit assumption of both styles of interaction has been that the human will be explicit, unambiguous and fully attentive. Equivocation, contradiction and preoccupation are unthinkable even though they are very human behaviors. Not allowed. We are expected to be disciplined, fully focused, single minded and 'there' with every attending muscle in our body. Worse, we accept it.

Times will change. Cipolla, Pentland *et al*, fly in the face (pun intended) of traditional human-computer interface research. The questions they pose and answers they provide have the common thread of concurrency. Namely, by combining modes of communication, the resulting richness of expression is not only far greater than the sum of the parts, but allows for one channel to disambiguate the other. Look. There's an example right there. Where? Well, you can't see it, because you cannot see me, where I am looking, what's around me. So the example is left to your imagination.

That's fine in literature and for well codified tasks. Works for making plane reservations, buying and selling stocks and, think of it, almost everything we do with computers today. But this kind of categorical computing is crummy for design, debate and deliberation. It is really useless when the purpose of communication is to collect our own thoughts. Under such conditions what you say is often far less important than how you say it. Gesture and facial expression are signal, not noise as some might have them, and sometimes so powerful that words are incidental.

Your face is your display. This display is wired so tightly to what you say, it is almost impossible to turn it off. Watch somebody talking on