Towards Man-Machine Interfaces: Combining Top-down Constraints with Bottom-up Learning in Facial Analysis

Vinay P. Kumar

Towards Trainable Man-machine Interfaces: Combining Top-down Constraints with Bottom-up Learning in Facial Analysis

by

Vinay P. Kumar

B.Tech. in Electrical Engineering, Indian Institute of
Technology, Mumbai, 1994
M.S. in Electrical and Computer Engineering, Northeastern
University, Boston, 1996

Submitted to the Department of Brain and Cognitive
Sciences in partial fulfillment of the requirements for the
degree of

Doctor of Philosophy in Computational Cognitive Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2002

Certified by: Tomaso Poggio
Uncas and Helen Whitaker Professor
Thesis Supervisor

Accepted by: Earl K. Miller
Co-Chair, Department Graduate Committee

Towards Trainable Man-machine Interfaces: Combining Top-down Constraints with Bottom-up Learning in Facial Analysis

by

Vinay P. Kumar

Abstract

This thesis proposes a methodology for the design of man-machine interfaces by combining top-down and bottom-up processes in vision. From a computational perspective, we propose that the scientific-cognitive question of combining top-down and bottom-up knowledge is similar to the engineering question of labeling a training set in a supervised learning problem.

We investigate these questions in the realm of facial analysis. We propose the use of a linear morphable model (LMM) for representing top-down structure and use it to model various facial variations such as mouth shapes and expression, the pose of faces and visual speech (visemes). We apply a supervised learning method based on support vector machine (SVM) regression for estimating the parameters of LMMs directly from pixel-based representations of faces. We combine these methods for designing new, more self-contained systems for recognizing facial expressions, estimating facial pose and for recognizing visemes.

Thesis Supervisor: Tomaso Poggio
Title: Uncas and Helen Whitaker Professor

Contents

3

Acknowledgments

The six years, during which I worked on the research reported here, have also been a period of drastic self-transformation. It has shaken my world-view to the very roots and left me a more mature and hopefully better person. To the extent that every single person who came into my life played a role in this transformation, and to the extent that this transformation has influenced my research, the list of people to thank is endless. In the interests of space, I shall limit myself to thanking those whose contribution to this thesis has been more direct.

I must first thank my parents who have lived with immense grief on my account, and yet never lost hope in me and kept wishing the best for me. I shall forever be grateful to their prayers on my behalf. At MIT, Tommy was a great advisor, providing the kind of support, encouragement and freedom that few graduate students can hope for. I sincerely thank him for his stewardship of this PhD program.

Many people helped me at various stages of this thesis in terms of code and discussions. Mike Jones gave me the invaluable code for building morphable models, Shayan Mukherjee was always patient with my many questions on support vector machines and Mike Oren taught me about wavelets. Thomas Vetter, Edgar Osuna, Volker Blanz, Antonio Torralba, Chikahito Nakajima, Martin Szummer, Ryan Rifkin, Bernd Heisele and Kah Kay Sung were very helpful when I needed help. Marypat and Gadi, in their own inimitable ways made my stay at CBCL smooth and lively.

Finally, I made some very special friends in BCS and CBCL who made a great difference, not only to my work in and out of the department, but also to my growth as a person. Siddharth taught me how to be a good idealist, Rajesh how to be a good realist and Tony how to actually do things.

Chapter 1

Introduction

In recent years, the problem of detecting and analyzing faces has become a core problem of computational vision. This problem has proved to be challenging not only from the perspective of the cognitive scientist who seeks to understand the mechanisms underlying perceptual tasks but also from the point of view of an engineer who seeks to design systems for the detection and analysis of faces. In this thesis, we study and compare the roles of top-down and bottom-up processes in facial analysis. In the process, we seek to address a question in cognitive science about the role of top-down information in perceptual tasks and also provide a design for a system that can be trained for facial analysis.

The scientific-biological question of perception has been addressed in the theory of computational vision. As propounded by Marr [33], this theory views vision as a purely bottom-up (feed-forward) process where an input representation of the image is converted in stages to a full-fledged 3D representation. All domain-specific knowledge is viewed as being hard-coded in the architecture of the bottom-up process. However, right from the time of Gestalt psychologists (Kohler [29]) it has been shown that a priori organizational principles might play a crucial role in even very simple perceptual experiences. These principles were supposed to be neurally encoded and mostly unconscious. Later, Bruner [7] showed that perception may be guided by conscious beliefs and expectations. A great deal of psycho-physical results have been adduced in support of these claims. However, this influence of unconscious organizational principles or conscious beliefs and expectations, often loosely called top-down influences has not been well-understood from a computational perspective with regards to its representation

and its interaction with bottom-up processes.

Recent studies (Cavanagh [11], Mumford [35], Jones, Sinha, et al. [34]) have analyzed aspects of top-down influences in vision from a computational perspective. While these studies have been motivated by scientific-biological questions, they have relied on models from machine vision for representing top-down information. In Cavanagh [11], top-down information is represented as prototypes of object classes stored in memory which are used to match input images. In Jones, Sinha, et al. [34], a morphable model (Beymer and Poggio [3], Vetter and Poggio [52], Jones and Poggio [28]) represents top-down information. The latter work also explores the implications of top-down influences on immunity to noise and missing input. In both these works, the issue of the interaction of top-down influences on bottom-up processes has been answered by invoking the scenario of analysis by synthesis. Mumford's [35] Pattern Theory also relies on analysis by synthesis for its explanation of perception. However, such total reliance on an analysis by synthesis approach presents two problems:

- Analysis by synthesis in general relies on a trivial bottom-up process i.e. one where there is no transformation of the input signal. However, we do know that the input (retinal) image in an organism undergoes some complex transformations. This raises the question: why does the input image undergo a complex transformation and what relation could this transformation have to any top-down information? In general, we should expect the transformations in the bottom-up process to be affected by the representation of top-down information and vice-versa. This aspect of the problem is not addressed in the analysis by synthesis scenario.

- Analysis by synthesis in general works only through iterative procedures. It relies on a search in the space of parameters representing the top-down "model" to find the best parameters which minimize some form of error measure between the model image and the input image. In practice, this is achieved through some kind of gradient-based search which is computationally intensive and seems less plausible given the complexity involved in the task of perceptual recognition.

Unlike the top-down processes, the bottom-up processes in perception have seen extensive investigation from a computational perspective and application in various engineering systems. Several vision problems ranging from the low and medium level ones such as edge detection,

depth estimation and optical flow to the high level ones such as structure from motion and object recognition, have been cast in a bottom-up manner and algorithms suggested for their implementation. The framework of learning theory has proved particularly successful for modeling certain phenomenon such as object detection as a bottom-up process. Statistical Learning theory was developed by Vapnik and co-workers [50, 49, 51] as the problem of estimating input-output relationships from empirical data. Independently, the theory of regularization developed by Tikhonov and Arsenin [43] was applied to a learning problem, namely estimating neural network architectures, by Girosi, Jones and Poggio [20]. This work found application in the detection of faces (Sung and Poggio [40], Rowley, Baluja, et al. [23]), pedestrians and cars (Papageorgiou et al. [9]). In these approaches top down information was incorporated through a labeled training set and appears as part of the training phase only. Whether top-down influences have any further structure and if they play any role in the actual perceptual task is not clear.

The study of the top-down and bottom-up processes is relevant not merely for the scientific understanding of perception but also for the design of technologies that mimic human perceptual capacities, and thus has application in a variety of man-machine interfaces. As a general rule, most systems that seek to mimic human perception, do not function without recourse to any constraints, but instead rely on a variety of constraints that encode physical, biological and ecological knowledge. For the task of higher-level perception, physical constraints are insufficient and biological and ecological constraints are hard to represent. Therefore, systems such as the ones performing object recognition, increasingly rely on manual annotation as a means of inserting top-down knowledge. Any step taken towards a general and structured representation of biological and ecological constraints inherent in the human perceptual process would make the task of designing these systems significantly easier.

In this thesis, we seek to view top-down influences in a more structured manner and provide a specific role for top-down information in the bottom-up process. In particular, the bottom-up process is cast in the framework of learning where the goal is to learn a mapping from an input space to an output space. The input space is typically the space of sensory data or observables and the output space represents the categories experienced in perception. The bottom-up process is represented by the function obtained by learning, that maps the sensory data to the the perceptual categories. We suggest that top-down information should specify a structure on the output space or the space
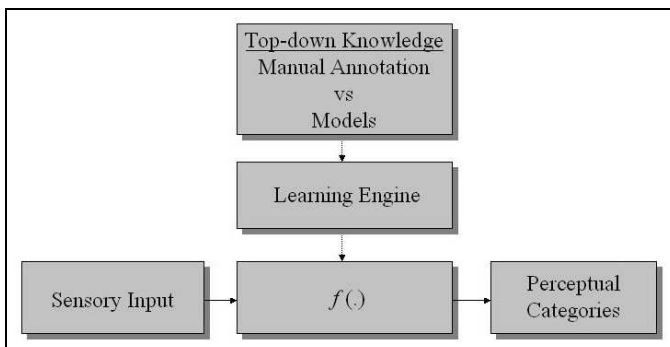
Figure 1.1: A block design for a system that mimics perceptual ability. The role of top-down knowledge is inherent in manual annotation and it can conceivably be replaced by structures that model the space of perceptual categories.

of perceptual categories, or in other words be a model of the space of perceptual categories. Having such a structure would allow us to replace the manual annotation of examples with parameters of the structure that models the output space. This way of providing top-down information allows us to make the design of man-machine interfaces much more self-contained, requiring less manual input. In Fig. 1.1 we illustrate this scheme in the form of a block diagram.

We demonstrate that such a self-contained man-machine interface system is possible by adopting a two-pronged approach. On the one hand, we show that the framework of learning is suitable for modeling the bottom-up process. We have demonstrated this by the use of a learning-based approach in the analysis of mouths (Kumar and Poggio [47]). In this, the learning approach (earlier characterized by the detection problem where a discrete object class was estimated) was extended to the problem of estimating continuous quantities such as openness and smile of mouths. This work was motivated by the goal of designing a real-time system capable of locating and analyzing human faces for expressions. It works on the principle of learning a regression function from a Haar wavelet based input representation of mouths to hand labeled parameters denoting openness and smile.

However, this work also opens up the possibility of being able to learn parameters of models that are capable of modeling the class of mouths, if such models exist. We consider a linear morphable model as an appropriate structure for the output space of the learning problem for estimating facial expression. We learn regression functions that

map pixel-based representations of images into the parameters of a morphable model. We show that this has applications in different aspects of man-machine interfaces including expression recognition, viseme recognition and pose estimation. In the next two sections, a brief background on these approaches is provided.

1.1 Learning for Bottom-up Analysis: A Real-time System for Facial Analysis

A system capable of locating human faces and analyzing and estimating the expressions therein in real time has applications in intelligent man-machine interfaces and in other domains such as very low bandwidth video conferencing, virtual actor and video email. The problem is difficult because it involves a series of difficult tasks each of which must be sufficiently robust. The main tasks in such a system can be summarized as follows.

1. Detect and localize faces in a cluttered background.

2. Detect and localize different facial features such as eyes, nostrils and mouth.

3. Analyze these regions to estimate suitable parameters.

The key theoretical issues that need to be addressed to facilitate such a real-time system are the image representations used for estimation of different features and parameters and the algorithms used for their estimation.

Facial patterns and their changes can convey a wide range of expressions, and regulate spoken conversation and social interaction. Faces are highly dynamic patterns which undergo many non-rigid transformations. Much of the previous work to capture facial deformations has relied heavily on two forms of parameterizable models, namely, geometry of face musculature and head shape (Blake and Isard [24], Essa and Pentland [18], Terzopoulos and Waters [42], Yuille et al. [1]) and motion estimation (Black and Yacoob [5], Ezzat [19]). While the former needs to be hand-crafted, the latter relies on repeated estimation of optical flow which can be computationally intensive. An important aspect of our work is that the models for all of the sub-tasks needed to achieve the final goal is automatically learned from examples. Moreover it does not make use of any 3D information or explicit motion or segmentation. Some work on 2D view-based models of faces have

also been extended to expression recognition (Beymer et al. [4], Beymer and Poggio [3] and Cootes et al. [12]). However these view-based models are explicitly dependent on correspondence between facial features and therefore repeated estimation of this correspondence during analysis. In designing our system, we built models that do not rely on such explicit correspondence.

The analysis of faces within a view-based paradigm is achieved by the localization and analysis of various facial regions. This is quite a hard problem due to the non-rigid transformations that various facial regions undergo. Some regions of the face (such as mouths) can vary widely. A pixel based representation for such a widely varying pattern is often inadequate. While localization of some facial regions may be simplified by the localization of the face, the task of mapping the facial region (e.g. the mouth) to a set of parameters (such as degree of openness or smile) is a daunting one since it is not clear how such a map can be made robust not only to the various factor affecting the image pattern of the facial region but also to errors in its localization.

The wavelet based representation has a long history and has recently been applied to the problems of image database retrieval in Jacobs et al. [26] and pedestrian detection in Papageorgiou et al. [9]. It has been shown to be robust to lighting conditions and variations in color. It is also capable of capturing in its overcomplete form the general shape of the object while ignoring the finer details. It is also known that at low enough resolutions it is tolerant to small translations. In our approach we have used an overcomplete Haar wavelet representation to detect nostrils to first localize the mouth and then use a sparse subset of coefficients of this overcomplete wavelet dictionary as a set of regressors which output the different parameters desired (in our case degree of openness and smile). This approach is similar to the method used by Graf et al. [22] and Ebihara et al. [15] where the outputs of different band-pass components of the input image are used as input to a classifier.

A face detection system is the first step towards any mouth analysis system. The face detection part was described in its non real-time form in Osuna, et al. [14]. This system is an example of how we address the other issue related to our problem, namely, of robust learning techniques for estimating the data-driven models. For this purpose, we used the Support Vector Machine (SVM) classification (Cortes and Vapnik [13]) which is based on the principle of structural risk minimization and thus minimizes a bound on the generalization error. In order to make this system real-time we have incorporated skin-segmentation to reduce the search space for the face detector. Localization of eyes

and nostrils is used to localize the mouth.

The problem of mapping the localized mouth region to the set of desired parameters is posed as the problem of learning a regression function from the Haar wavelet coefficients of the image to parameters of openness and smile obtained by manually annotating a training set. This problem is tackled using SVM regression and forms the most important component for our purposes. This system has been trained on multiple faces and is capable of detecting faces and localizing eyes, nostrils and mouths for multiple persons in a general environment. Furthermore, the system is trained to analyze the localized mouth pattern to estimate degrees of openness and smile. The estimated parameters are used to drive the head movements and facial expressions of a cartoon character in real time on a general purpose computer.

1.2   Learning Top-down Parameters

The design of a learning-based real-time system for tracking and analyzing faces shows that it is possible to use purely bottom-up methods to make head-way in some hard problems of perception. However, the fact that the system uses manually annotated examples for learning shows that top-down knowledge is inherent to its success. Is manual annotation the only way of specifying top-down knowledge? In the case of facial analysis, this question can be asked more pointedly as follows.

- What are the basic units of facial shapes/motions? How can they be represented?

- What relation does the pixel image of faces bear to these representations? Can those be learnt?

The answer to the first question demands a model for modeling the class of mouths which is capable of representing the inherent constraints on mouth shapes and motions. Amongst the many model-based approaches to modeling object classes, the Linear Morphable Model (LMM) is an important one. The antecedents of LMMs can be found in the work of Beymer and Poggio [3], Vetter and Poggio [52] who describe the modeling of different objects of the same object class using a linear combination of prototypes, and the work of Ullman and Basri [45] who model the multiple views of a single object by linear combination of prototypical views. A LMM works by establishing pixel-wise correspondence between the prototypical images of an object class. This leads to a dual-representation of an image as a texture and a shape.

By linearly combining the texture and shape separately, it is possible to generate new images of objects in the same class. Mathematically a LMM represents a manifold in the space of pixels on which any image belonging to the class is constrained to lie. LMMs will be explained in greater detail in a later chapter.

LMMs have been used to model object classes such as faces, cars and digits (Jones and Poggio [28], Blanz [46], Cootes et al. [12]). Recently it has been used for the synthesis of visual speech (Ezzat and Poggio [41]). But so far, it has mainly been a tool for image synthesis and its use for image analysis has been somewhat limited, mainly for the purpose of verification after the stage of object detection. Such analysis has been approached through the computationally intensive analysis by synthesis method only. In Jones and Poggio [28] and Cootes et al. [12], the matching parameters are computed by minimizing the squared error between the novel image and the model image using a stochastic gradient descent (SGD) algorithm. This technique is computationally intensive (taking minutes to match even a single image). A technique that could compute the matching parameters with considerably less computations and using only view-based representations would make these models useful in real-time applications.

In this thesis, we explore the possibility of learning to estimate the matching parameters of a LMM directly from pixel-based representations of an input image. LMMs have the good property that by virtue of construction (using examples) a LMM represents the statistics of the class of images as opposed to other models which need to be hand-crafted (e.g. those based on facial musculature). What is being investigated here is the possibility that *the statistics represented by a LMM forms the basis of top-down information, at least in the earliest stage of perception.* Thus it may be natural to learn the parameters of a LMM as the first stage of a perceptual task.

The work of Cootes and co-workers [12] has come close to approximating our methods. Their model for object classes known as the active shape model is similar in spirit to the LMM but differs in its details. It lacks the dense correspondence field of the LMM. However, they attempt to compute the matching parameters of their active shape model from the image with a view to speed-up the process of analysis by synthesis. The speed-up is achieved by learning several multivariate linear regressions from the error image (difference between the novel and the model images) and the appropriate perturbation of the model parameters (the known displacements of the model parameters), thus avoiding the computation of gradients. This method is akin to learning the tangent plane to the manifold in pixel space formed by the mor-

phable model. In our method, the emphasis is on learning the complex non-linear transformations that would carry the input representation to the matching parameters of the LMM directly.

Since LMMs are constructed from examples, based on the kind of examples used, it is possible to define qualitatively, two kinds of LMMs. The first kind of LMMs model the space of line-drawings and need to be built using examples of line drawings. This can also be interpreted as being constructed with phenomenal examples and modeling a phenomenal space. The other kind of LMMs are built using real images and therefore model the space of pixels. As a result one can conceive of two different ways in which LMMs can form the basis of top-down knowledge in perceptual systems. When using line-drawing LMMs, the LMMs represent phenomenal categories and therefore by learning the mapping from pixel-based representations to LMM parameters we are learning to directly estimate phenomenal categories. The same is not true of the pixel LMMs and extracting meaningful phenomenal categories might require further analysis. These issues will be discussed in this thesis.

In this thesis, we propose to construct pixel LMMs to model various mouth shapes and expressions. Following Jones and Poggio [28], the LMM is constructed from examples of mouths. Principal Component Analysis (PCA) on the example textures and flows allows us to reduce the set of parameters. We then use SVM regression (Vapnik [51]) to learn a non-linear regression function from a sparse subset of Haar wavelet coefficients to the matching parameters of this LMM directly. The training set (in particular the y of the (x, y) pairs) for this learning problem is generated by estimating the true matching parameters using the SGD algorithm described in Jones and Poggio [28].

We explore different aspects of this method. Clearly the performance is likely to be affected by the number of different people the LMM tries to model. Accordingly we consider the case of a single person LMM and a multiple person LMM. An obvious extension to estimating LMM parameters directly is to initialize a subsequent gradient descent algorithm for the more accurate estimation of the parameters. We present some results on the kind of speed up that is likely with this approach.

1.3   Applications to Man-Machine Interfaces

Estimation of top-down LMM parameters could be applied to the design of intelligent systems in two different ways.
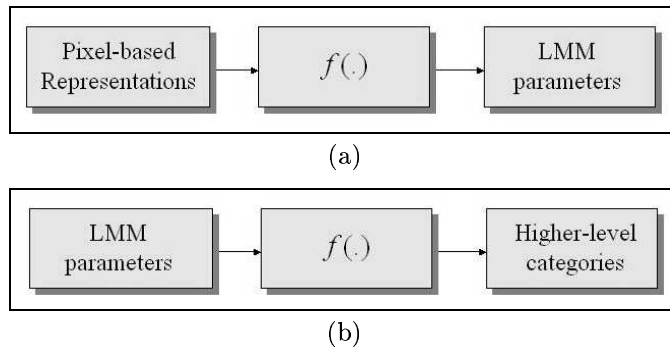
Figure 1.2: Illustrating the two kinds of applications that the estimation of the LMM parameters can be used for. (a) The LMM parameters constitute the higher-level categories and are estimated from pixel-based representations and (b) the LMM parameters are used as image features and the higher-level categories are specified separately.

- In one case, higher-level perceptual categories are intrinsically represented by the LMM parameters. It may require some post-processing to extract these categories but *no additional learning is required*. Therefore, the basic framework of these systems is to map from pixel-based representations to the higher level categories encoded in the LMM parameters (see Fig. 1.2(a)). It is this case which makes possible the design of self-contained man-machine interfaces.

- A different situation arises, when we consider using the LMM parameters as a feature vector for the image. In this case the LMM parameters do not represent the perceptual categories of interest, but serve as input to an additional stage of learning, in which the final category to be estimated may be hand-labeled (see Fig. 1.2(b)).

We demonstrate the former concept of a self-contained man-machine interface, through a new system that can track facial expressions. The system works by estimating the parameters of a pixel LMM of the mouth shapes of a single person using SVM regression. It is obvious that this would allow us to track mouth movements. Additionally, we also show that the principal components of the flow space of the mouth LMM correspond to distinct facial expressions. Therefore estimating LMM parameters allows us to track facial expressions too. In contrast to all earlier systems tracking facial expressions (Yuille et al. [1], Ter-

zopoulos and Waters [42], Blake and Isard [24], Essa and Pentland [18], Black and Yacoob [5], Beymer et al. [4], Ezzat [19], Cootes et al. [12] and Kumar and Poggio [47]) our system is able to directly estimate top-down (LMM) parameters from bottom-up pixel-based representations, and thus avoids manual annotation.

Continuing in a similar vein, we also explore the possibility of learning to estimate the parameters of a line drawing LMM. We construct a line drawing LMM that models facial pose and SVM regression and we learn to estimate its parameters from pixel-based representations of the image. The training set in this case is obtained by matching a coarse edge map of the input image to the line drawing LMM using the SGD algorithm. Facial pose estimation has always been important for subsequent facial analysis including face recognition. Much previous work on pose estimation has relied on feature detection followed by geometric modeling to estimate the pose. Some work has been done on matching models (morphable models in case of Beymer and Poggio [2], elastic graphs in case of Kruger et al. [30]) using analysis by synthesis methods. Recently it has been shown by Sherrah et al. [25] that Gabor wavelets combined with PCA on pixel images can yield a pose-similarity space. Our work carries this idea forward and we compute relationships between pixel representations and a pose space defined using a line-drawing LMM.

Recently it has been suggested that LMM parameters could be used for higher level analysis such as face identification (Blanz [46] and Edwards et al. [21]). These methods fall in the latter category of applications i.e. based on supervised learning with the LMM parameters as an image feature. The application we consider is the recognition of visemes. Visemes are the visual analogues of phonemes (Ezzat and Poggio [41]). Recognizing visemes have potential applications in enhancing the performance of speech recognition systems or driving photorealistic avatars. A full treatment of visemes is beyond the scope of this thesis. However we test the simple technique of mapping single images into viseme classes by training classifiers on the matching LMM parameters. Although this method has not been fully explored, it raises questions on the ability of LMM parameters to substitute for image features.

1.4   Significant Contributions

The significant contributions of this thesis extend to both, questions of a scientific-biological nature as well as of an engineering kind.

The significant contributions of this thesis to engineering applica-

tions are

- A purely bottom-up and data-driven learning-based real-time system capable of tracking faces and analyzing for basic expressions such as openness and smile, as well as pose.

- Modeling mouth shapes and expressions using pixel LMMs and facial pose using line-drawing LMMs.

- A learning-based approach to estimate the parameters of LMMs directly from the image.

- Combining the above approaches and their application to the design of new systems with vision capabilities such as expression recognition, pose estimation and viseme recognition.

The main contribution of this thesis to scientific-biological questions lies in the suggestion of a specific structure and role to top-down information in bottom-up processes for perceptual tasks - namely that a morphable model is capable of representing top-down knowledge and that the purpose of learning is to establish a (direct) relationship between the incoming retinal image and the morphable model space.

1.5   Overview of the Thesis

In the next chapter, we describe in detail the different components of the real-time system for face-tracking and analysis. We bring out the bottom-up, data-driven nature of the system and compare its performance with other systems with a similar goal. In chapter 3, we shall discuss the design of a morphable model for the class of mouth images and learning its parameters directly from the image. In chapter 4, we describe various applications of our approach which include expression recognition, pose estimation and viseme recognition. Finally we conclude by considering the new questions that this work raises and its implications for cognitive science as well as engineering.

Chapter 2

Learning-based approach to Real Time Tracking and Analysis of Faces

This chapter describes a trainable system capable of tracking faces and facial features like eyes and nostrils and estimating basic mouth features such as degrees of openness and smile in real time. In developing this system, we have addressed the twin issues of image representation and algorithms for learning. We have used the Haar wavelet representation to robustly capture various facial features. This system, unlike previous approaches, is entirely trained using examples and does not rely on a priori (hand-crafted) models of facial features based on optical flow or facial musculature. Therefore it represents a purely bottom-up (feed-forward) approach to facial analysis.

The system works in several stages that begin with face detection, followed by localization of facial features and estimation of mouth parameters. Each of these stages is formulated as a problem in supervised learning from examples. We apply the new and robust technique of support vector machines (SVM) for classification in the stage of skin segmentation, face detection and eye detection. Estimation of mouth parameters is modeled as a regression from a sparse subset of coefficients (basis functions) of an overcomplete dictionary of Haar wavelets.

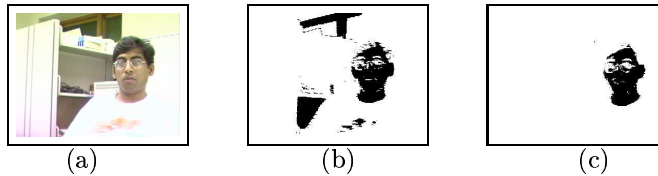<center>(a)   (b)   (c)</center>

Figure 2.1: Illustrating the use of the skin segmentation and component tracking. (a) The indoor scene, (b) after skin segmentation, (c) after component extraction and tracking.

2.1 Face Detection System

In this section, we briefly describe the face detection system which localizes a frontal and unoccluded face in an image sequence and tracks it. It uses skin segmentation and motion tracking to keep track of candidate regions in the image. This is followed by classification of the candidate regions into face and non-face thus localizing the position and scale of the frontal face.

2.1.1 Skin segmentation and component tracking

Skin detection is used to segment images into candidate head regions and background. The skin detector works by scanning the input image in raster scan and classifying each pixel into skin or non-skin. The skin model is obtained by training a SVM using the two component feature vector $(\frac{r}{r+g+b}, \frac{g}{r+g+b})$ where (r, g, b) are red, green and blue components of the pixel. There exists more sophisticated features for skin detection (Lee and Goodwin [31]) but due to constraints of real-time processing we used a simple one. We collected over 2000 skin samples of different people with widely varying skin tone and under differing lighting conditions.

The skin segmented image is analyzed into connected components. We encode and keep track of the positions and velocities of the components. This information is used to predict where the component will be seen in the next frame and thus helps constrain our search for skin. Components that are smaller than a predefined threshold or those that have no motion at all are discarded from consideration. As time goes on (in our case after 20 frames after a complete rescan), the skin search is further restricted only to regions with active components (i.e. with faces). As a result the performance of this stage actually improves since spurious skin is eliminated and the search is heavily constrained

in spatial extent as illustrated in figure 2.1.

Both skin segmentation and component analysis is performed on a sub-sampled and filtered image and is therefore fast and reliable. This stage has proven to be very useful since it eliminates large regions of the image and several scales (image resizing being computationally intensive this is particularly useful) from consideration for face detection and helps in doubling the frame rate.

2.1.2 Face Detection

The SVM classifier used in the face detection system has been trained using 5,000 frontal face and 45,000 non-face patterns, each pattern being normalized to a size of 19×19. It compensates for certain sources of image variations by subtracting a best-fit brightness plane from the pixel values to correct for shadows and histogram equalization of the image patterns to compensate for lighting and camera variations.

Face detection can be summarized as follows. For greater detail the reader is referred to Osuna, et al. [14].

- Consider each of the active components in sequence.

- Scale each component several times.

- Cut 19×19 windows from the scaled components.

- Pre-process each window with brightness correction and histogram equalization.

- Classify each window as either face or non-face

We take the very first scale and the very first location where a face is detected as the position of the face. We also keep track of the position and velocities of the faces within each component. This helps us in predicting the position of the face in the next frame and thus reduces our search further. Once the face is detected it is resized to a fixed size of 120×120 for further processing. The real-time face detection system works close to 25 frames per second.

2.2 Facial Feature Detection and Mouth Localization

Face localization can only approximately predict the location of the mouth. Since the face detection system is somewhat tolerant to rotations and tilting of the head, a mouth localizer that relies solely on

information of face location can be significantly thrown off the mark by head movements that are not in the plane of the image. Moreover, as mentioned earlier mouth shapes can change dramatically as the person's expression changes. Often it is not clear what is the ideal localization for such a widely varying pattern. All of the above phenomena can have adverse effects on the performance of the next stage of mouth analysis. Some mouth localizers exist in literature they either rely on color information (Oliver et al. [36]) or on optical flow (Black and Yacoob [5]) but our approach was to ensure that the mouth region remains stationary with respect to other more stable landmarks on the face such as eyes and nostrils. In this as well as the later stage of mouth analysis, the ability of the wavelet transform to encode localized variations plays a crucial role.

2.2.1 Encoding localized variations using Haar wavelets

Wavelets can be interpreted as encoding image variations locally in space and spatial frequency. In figure 2.2 we depict the 3 types of 2 dimensional Haar wavelets. These types include basis functions that capture change in intensity in the horizontal direction, the vertical direction and the diagonals (or corners). The standard Haar wavelets (Mallat [32]) are what are known as complete (or orthonormal) transform and are not very useful for our application. Since we need greater resolution, we implement an overcomplete (redundant) version of the Haar wavelets in which the distance between the wavelet coefficients at scale n is $\frac{1}{4}2^n$ (quadruple density) instead of 2^n for the standard transform. Since the spacing between the coefficients is still exponential in the scale n we avoid an explosion in the number of coefficients and maintain the complexity of the algorithm at $O(N)$ in the number of pixels N.

It is not unreasonable to expect the wavelet representation to be reliable for locating different facial features. Most areas of the face with high frequency content (strong variations) are localized around the eyes, the nose and the mouth. Thus they are amenable to detection by any method that encodes such local variations like the Haar wavelets. Figure 2.3 shows that in the Haar wavelet response to regions around the nose and the mouth, for the corner type basis, the nostrils and the contours of the mouth stand out (have a large absolute value for the response). These ideas have motivated us to use wavelet features as follows.

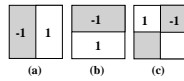1. For detection of nostrils.

Figure 2.2: The 3 types of 2 dimensional non-standard Haar wavelets;
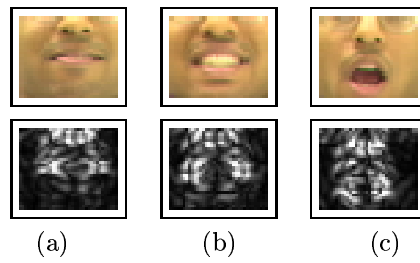(a) "vertical", (b) "horizontal" and (c) "corner".

Figure 2.3: Illustrating the corner type Haar wavelet response to 3
generic mouth shapes. Note that the nostrils and the contours of the
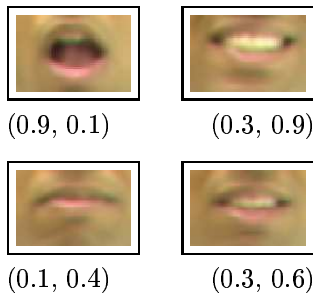mouth stand out.

(0.9, 0.1)	(0.3, 0.9)
(0.1, 0.4)	(0.3, 0.6)

Figure 2.4: Illustrating subjective annotation of the training set for degrees of (openness, smile).

2. As features for analysis of mouth region.

2.2.2 Locating eyes and nostrils

Slightly different procedures are applied to detect the eyes and nostrils. Eye detection is done by training a SVM classifier with more than 3000 eye patterns and 3000 non-eye patterns of size 13×9. In the interests of real time processing we train a linear classifier which takes normalized pixels as input. During detection several locations and scales within a candidate region is examined for the location of the eye. This admittedly leads to numerous false positives.

In order to eliminate false positives one can employ the following reasoning. From the set of positives, one can choose that positive with the highest likelihood of being the true positive. Let $\mathbf{x}$ denote the input feature vector and $y(\mathbf{x})$ the output of the SVM classifier. Thus $\mathbf{x} \in S_p$, the positive class if $y(\mathbf{x}) > 0$. Let $\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_m\}$ be the inputs classified as positive. If all except one is a false positive, then we can eliminate the false positives by choosing

$$\mathbf{x}^* = \arg\max_k P(\mathbf{x}_k | y(\mathbf{x}_k) > 0) \tag{2.1}$$

This involves estimating the multidimensional density $P(\mathbf{x}|y(\mathbf{x}) > 0)$ which can be very ill-posed. In order to simplify matters, we assume (see Platt [38]),

$$P(\mathbf{x}|y(\mathbf{x}) > 0) \approx P(y(\mathbf{x})|y(\mathbf{x}) > 0) \approx P(y(\mathbf{x})|\mathbf{x} \in S_p) \tag{2.2}$$

We can be easily estimate $P(y(\mathbf{x})|\mathbf{x} \in S_p)$ by plotting an histogram of the SVM output for the positive class from the training data. This

distribution turns out to be bell-shaped. Hence we can eliminate false positives by accepting the one positive for which the SVM output is closest to the maxima of this conditional distribution. This simple technique has given good results for false positive elimination.

Nostrils are detected by identifying a candidate nose region from the location of the face and finding the local maxima of the absolute value of Haar wavelet coefficients in the right and the left half of this region. Here we compute the Haar wavelets for basis functions of support size 4×4 at quadruple density. The local maxima can be taken for either the vertical or the corner wavelets.

We smooth the locations of eyes and nostrils thus obtained by a moving average filter to reduce jitter and increase robustness and tracking.

2.3 Mouth pattern analysis

In this section, we describe the analysis of the mouth pattern obtained from the previous stage of localization to estimate basic parameters such as degree of openness and smile. We have attempted to model the problem of mapping the mouth pattern to a set of meaningful parameters as learning the regression function from an input space to output parameters.

2.3.1 Generating the training set

The training set for learning the regression function is learned as follows.

- The mouth localization system is used to localize the mouths of different persons as they make basic expressions such as surprise, joy and anger.

- The mouths grabbed are normalized to a size of 32×21 and annotated manually for the degree of openness and smile on a scale of 0 to 1 (see Beymer et al. [4] and Beymer and Poggio [3]). This is done based on the subjective interpretation of the user and no actual image parameters are measured. Figure 2.4 shows some examples of such a subjective annotation.

2.3.2 Automatic selection of a sparse subset of Haar Wavelet coefficients

The input space for the regression function is chosen to be a sparse subset of the overcomplete Haar wavelet coefficients described in the previous section. In the course of the expression some of the coefficients change significantly and others do not. In the framework of data compression, we would project the data on a subspace which decorrelates it (which is the same as Principle Component Analysis) and then choose those variables with the highest variance. However, this is not the same as choosing a sparse subset where our purpose is to completely avoid the computation of those Haar coefficients that are not useful in the regression.

Choosing a sparser set of coefficients in a regression problem has the added advantage of reducing the variance of the estimate, although bias may increase. Conventional approaches to doing this include subset selection and ridge regression. In order to obtain a sparse subset of the Haar coefficients, we choose those Haar coefficients with the maximum variance. This is motivated by the fact that coefficients with high variance are those that change the most due to changes in mouth shapes and are hence statistically significant for "measuring" this change. In this application, we chose the 12 coefficients with the highest variance. The variance of a sub-sampled set of the Haar coefficients is shown in table 2.1

0.02	0.22	0.23	0.30	0.14
0.61	1.95	2.27	3.03	4.57
0.20	0.13	0.59	0.62	0.65

(a)

0.06	3.18	0.99	0.54	0.30
0.18	2.38	1.14	5.58	7.70
0.33	0.65	1.88	3.12	1.38

(b)

0.01	1.28	1.30	0.43	0.69
0.35	0.89	0.69	2.29	7.22
0.20	1.08	0.78	1.44	1.33

(c)

Table 2.1: Normalized variance values for the Haar coefficients of mouth sequences (as they open and close). (a) Vertical, (b) Horizontal and (c) Corner types.

2.3.3 Linear Least Squares regression on the Haar coefficients

Once we have have obtained the sparse subset of coefficients, the next task is to learn the linear map from these coefficients to the output parameters. We implement a simple least squares estimate of the coefficients of the linear map. Let $\mathbf{v}^{(n)}$ be the vector of coefficients obtained after choosing the sparse subset of Haar wavelet coefficients, for the nth training sample in a training set with N samples. Then,

$$
\mathbf{V} = \begin{bmatrix} \mathbf{v}^{(1)T} \\ \mathbf{v}^{(2)T} \\ \vdots \\ \mathbf{v}^{(N)T} \end{bmatrix} \tag{2.3}
$$

is the matrix of all the coefficients of the training set. Also if $\mathbf{y} = (y^{(1)}, y^{(2)}, \ldots y^{(N)})^T$ is the vector of all the outputs assigned to the training set. If $\mathbf{a}$ is the vector of weights of the linear regression then the problem is to find the least squares solution to $\mathbf{Va} = \mathbf{y}$. The solution is clearly $\mathbf{a} = \mathbf{V}^\dagger \mathbf{y}$ where $\mathbf{V}^\dagger$ is the pseudo-inverse of $\mathbf{V}$. During testing, the value of the output parameter is given by $y = \mathbf{a}^T \mathbf{v}$ where $\mathbf{v}$ is the vector of coefficients from the sparse subset of Haar coefficients.

2.3.4 Support Vectors for Linear Regression

In this section, we sketch the ideas behind using SVM for learning regression functions (a more detailed description can be found in Golowich, et al. [48] and Vapnik [51]) and apply it for the simpler case of linear regression. Let $G = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, be the training set obtained by sampling, with noise, some unknown function $g(\mathbf{x})$. We are asked to determine a function f that approximates $g(\mathbf{x})$, based on the knowledge of G. The SVM considers approximating functions of the form:

$$
f(\mathbf{x}, \mathbf{c}) = \sum_{i=1}^D c_i \phi_i(\mathbf{x}) + b \tag{2.4}
$$

where the functions $\{\phi_i(\mathbf{x})\}_{i=1}^D$ are called *features*, and b and $\{c_i\}_{i=1}^D$ are coefficients that have to be estimated from the data. This form of approximation can be considered as an hyperplane in the D-dimensional feature space defined by the functions $\phi_i(\mathbf{x})$. The dimensionality of the feature space is not necessarily finite. The SVM distinguishes itself by

minimizing the following functional to estimate its parameters.

$$R(\mathbf{c}) = \frac{1}{N} \sum_{i=1}^{N} \mid y_i - f(\mathbf{x}_i, \mathbf{c}) \mid_\epsilon + \lambda \|\mathbf{c}\|^2 \qquad (2.5)$$

where λ is a constant and the following *robust* error function has been defined

$$\mid y_i - f(\mathbf{x}_i, \mathbf{c}) \mid_\epsilon = \max(\mid y_i - f(\mathbf{x}_i, \mathbf{c}) \mid - \epsilon, 0) \qquad (2.6)$$

Vapnik showed in [51] that the function that minimizes the functional in equation (2.5) depends on a finite number of parameters, and has the following form:

$$f(\mathbf{x}, \alpha, \alpha^*) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) K(\mathbf{x}, \mathbf{x}_i) + b, \qquad (2.7)$$

where $\alpha_i^* \alpha_i = 0$, $\alpha_i, \alpha_i^* \geq 0$ $i = 1, \ldots, N$, and $K(\mathbf{x}, \mathbf{y})$ is the so called *kernel* function, and describes the inner product in the D-dimensional feature space

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{D} \phi_i(\mathbf{x}) \phi_i(\mathbf{y})$$

In our case, since we are implementing a linear regression, the features take the following form $\phi_i(\mathbf{x}) = x_i$ (the ith component of $\mathbf{x}$) and $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$. Now it is easy to see that the linear relation between the sparse subset of Haar coefficients and the output parameter is given by $y = \mathbf{a}^T \mathbf{v} + b$ where $\mathbf{a} = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) \mathbf{v}^{(i)}$. Only a small subset of the $(\alpha_i^* - \alpha_i)$'s are different from zero, leading to a small number of support vectors. The main advantage accrued by using a SVM is that since it uses the robust error function given by equation (2.6), we obtain an estimate which is less sensitive to outliers.

2.3.5 Results of Mouth Parameter Estimation

The real time face detection and mouth localization system works at close to 10 frames per second. This system was used to collect 771 examples of mouth images with different degrees of openness and smile for a single person and manually annotated as described in section 2.3.1. The images were pre-processed with illumination correction. This set was used to learn a linear regression function from the sparse subset of Haar coefficients to the output parameters, using the Linear Least Squares and the SVM criteria. We smooth the output of the regression
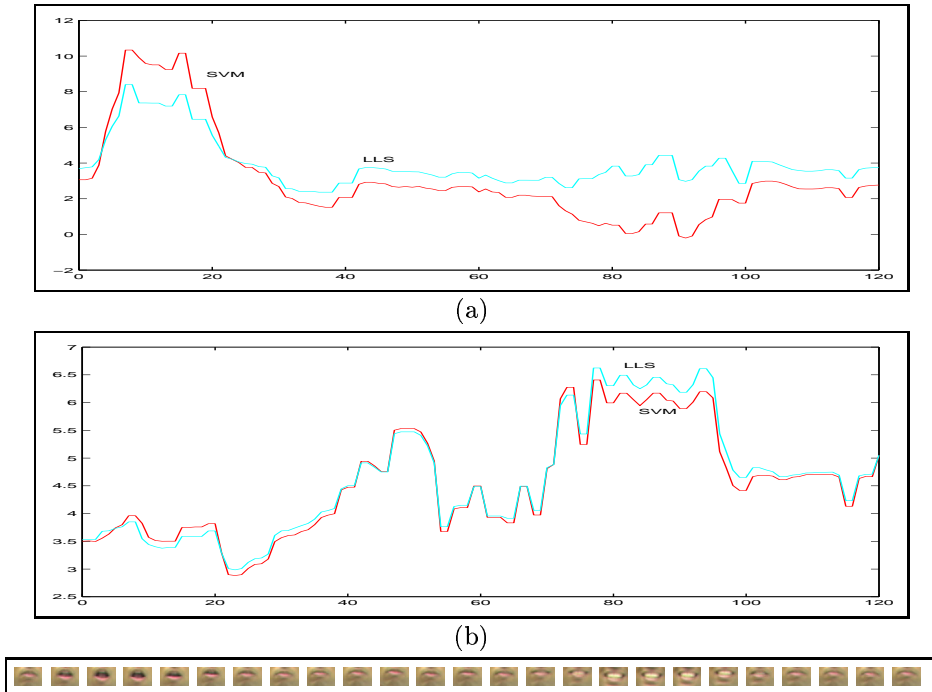
(a)

(b)

Figure 2.5: Estimated degree of (a) openness and (b) smile for a sequence of 121 images by linear regression learned using Linear Least Squares and Support Vector Machine with $\epsilon = 0.05$ and followed by median filtering with a filter of length 3. Higher values indicate a more open mouth and or a more smiling mouth.

by a median filter of length 3. In figure 2.5 we present the results of testing this regression technique on a test sequence of 121 images. One can note that the linear least squares estimate does as well as the SVM while estimating smile but performs poorly in the case of openness.

We have currently implemented the system to estimate the degree of openness and smile of mouths in real time. So far, the training set contains the images of only one person. But it shows a good capacity to generalize to the mouths of other people. This system also runs at 10 frames per second on a general purpose PC. We expect that when mouths of more people are added to the training set and a non-linear regression is implemented, it will be able to generalize even better.
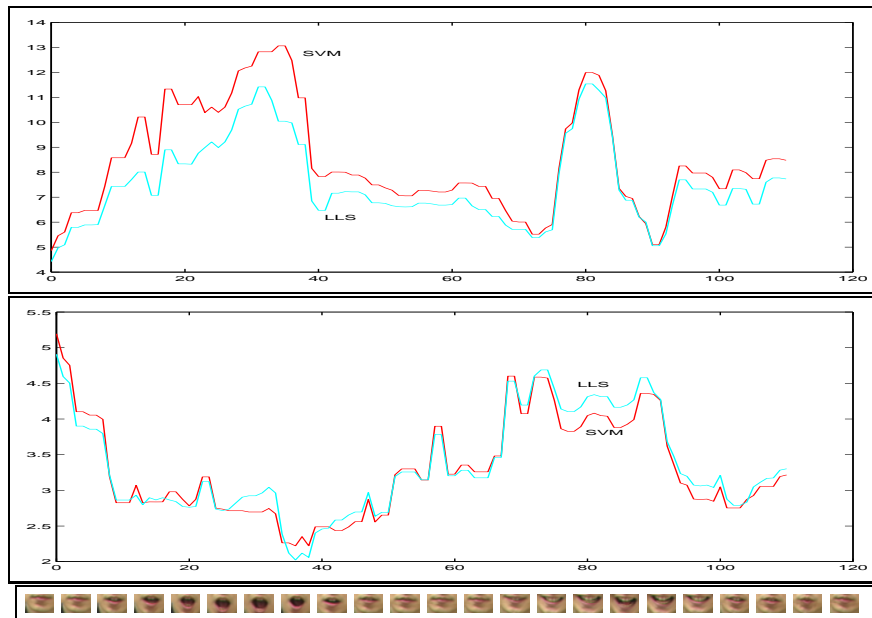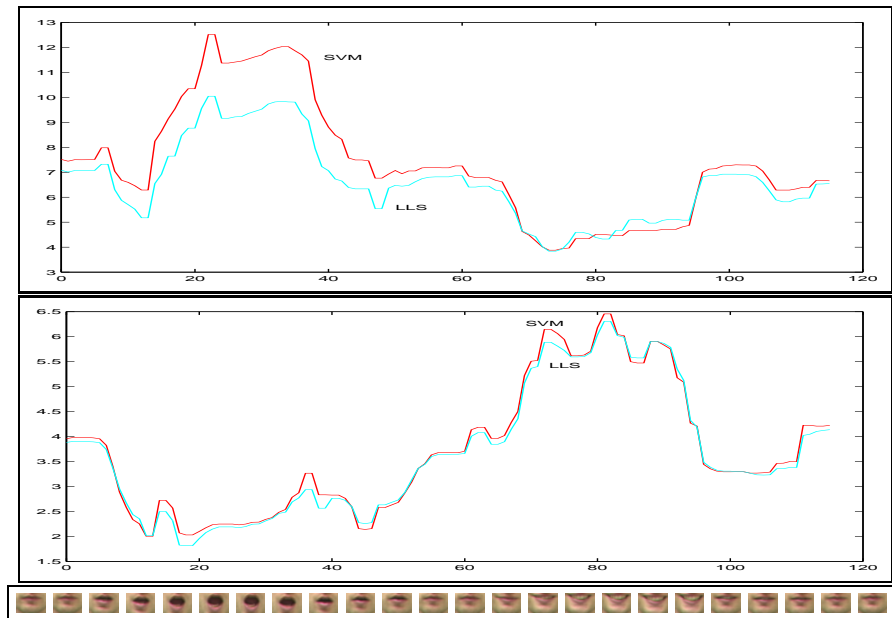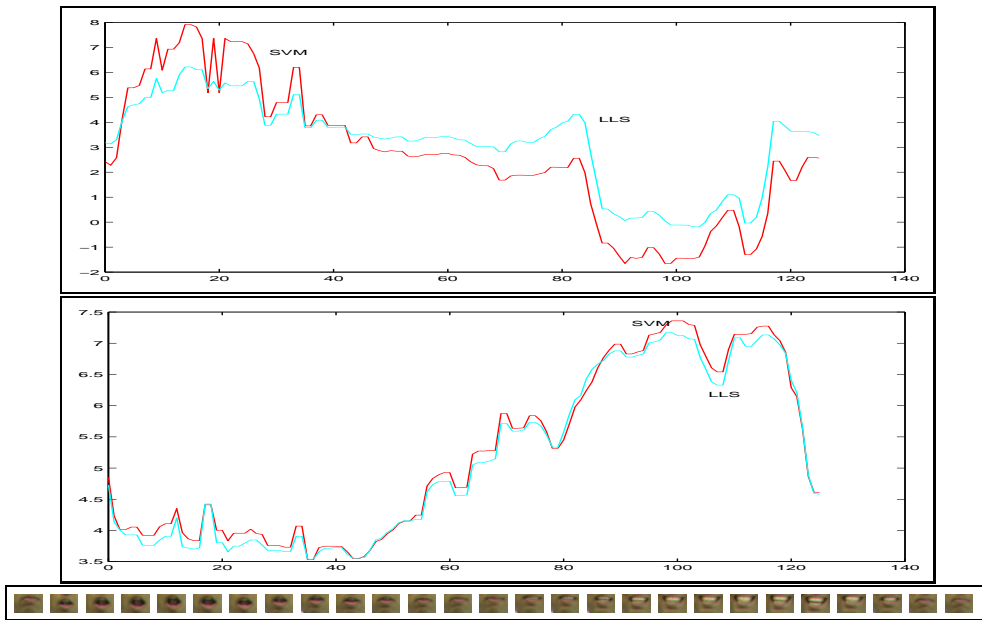
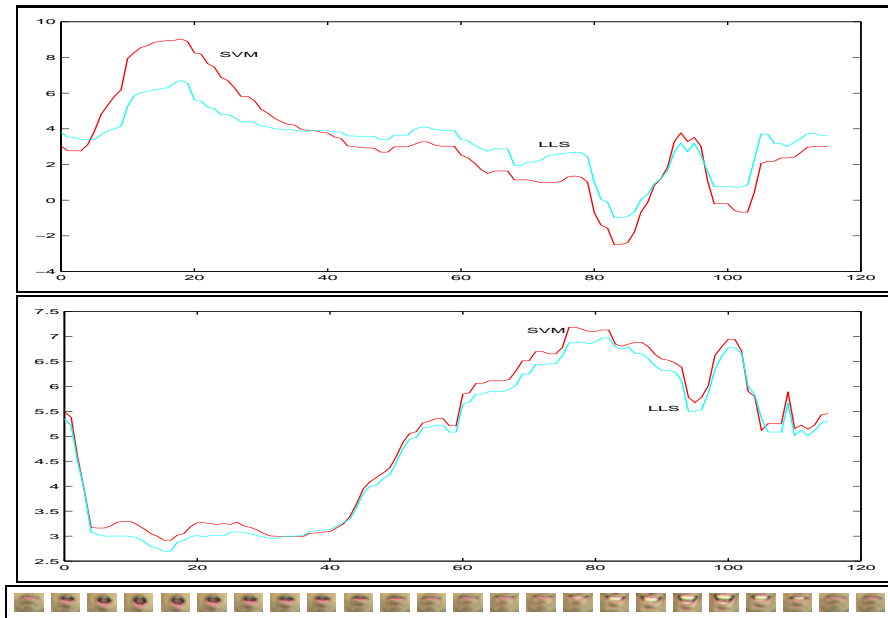Figure 2.6: Illustrating the ability of the mouth parameter estimator to generalize to different people.

Figure 2.7: Illustrating the ability of the mouth parameter estimator to generalize to different lighting conditions.

2.4   Discussion

In this chapter we described the application of wavelet-based image representations, and SVM classification and regression algorithms to the problem of face detection, facial feature detection and mouth pattern analysis. The basic motivation to use the above two techniques comes from the inherent robust characteristics of both: the ability of the former to represent basic object shapes and reject spurious detail and that of the latter to bound generalization error rather than empirical error.

In summary, in our approach, we use skin segmentation and component analysis to speed up a SVM classifier based face detector. We then detect stable features on the face such as eyes and nostrils which are used to localize the position of the mouth. After localization, we learn a linear regression function from a sparse subset of Haar coefficients of the localized mouth to outputs which represent the degree of openness and smile of the mouth. We have used the output of this system to drive the expressions and head movements of a cartoon character. We believe that the system can be improved by adding examples of facial expression of more people.

So far our technique has relied on purely bottom-up (feed-forward) processes to detect and analyze faces. But the fact that a manually annotated training set was used in training the system suggests that key top-down knowledge had to be incorporated in order to make the system perceptually meaningful. Clearly, if the training set was randomly labeled, the system's output would have no perceptual meaning. This then prompts an open question: What are the ways to incorporate top-down knowledge that would "specify" what parameters need to be estimated by the bottom-up processes described in this chapter? In the next chapter we consider this question in greater detail.

Chapter 3

Learning-Based Approach to Estimation of Morphable Model Parameters

In Jones, Sinha, et al. [34], a morphable model was suggested as the vehicle for embodying top-down knowledge. If so, then it follows that a morphable model should constrain the bottom-up learning process which converts pixel information into higher-level cognitive categories. One simple way of achieving this is to have the output of the learnt function in the space of the morphable model representing the top-down information. In this chapter we describe a method for performing the same. This method uses a learning-based approach to estimate the LMM parameters directly from the images of the object class (in this case mouths).

The motivation for this work comes from the use of a learning-based approach in real-time analysis of mouths (Kumar and Poggio [47]), in which it was shown that a regression function can be learnt from a Haar wavelet based input representation of mouths to hand labeled parameters denoting openness and smile. Here we extend the method to learn to directly estimate the matching parameters of an LMM from images.

This method can be used to bypass or speed up current computationally intensive methods that use analysis by synthesis, for matching objects to morphable models. We represent the mouth images as a
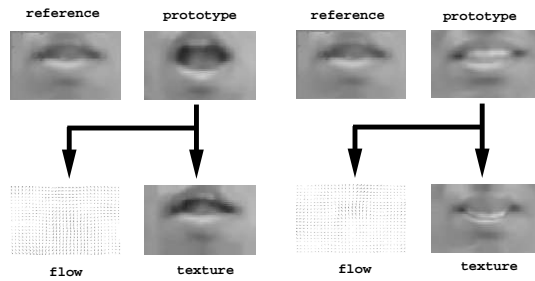
Figure 3.1: Illustrating the vectorized representation of an image. Pixel-wise correspondences are computed between the prototype image and a standard reference image. The flow vector consists of the displacements of each pixel in the reference image relative to its position in the prototype. The texture vector consists of the prototype *backward warped* to the reference image.

sparse subset of low resolution Haar wavelet coefficients and apply the robust technique of Support Vector Machines (SVM) for learning a regression function from the Haar coefficients to the LMM parameters. We study the speed up that might be obtained by estimating these parameters using a regression function to initialize, as opposed to standard analysis by synthesis.

3.1 Linear morphable model for modeling mouths

In this section we provide a brief overview of LMMs and their application to modeling mouths.

3.1.1 Overview of LMMs

A linear morphable model is based on linear combinations of a specific representation of example images. The representation involves establishing a correspondence between each example image and a reference image. Thus it associates with every image a shape vector and a texture vector. Figure 3.1 illustrates this vectorized representation, which can be computed by the linear combination of example images as shown in Figure 3.2 (See Jones and Poggio [28] for more details).
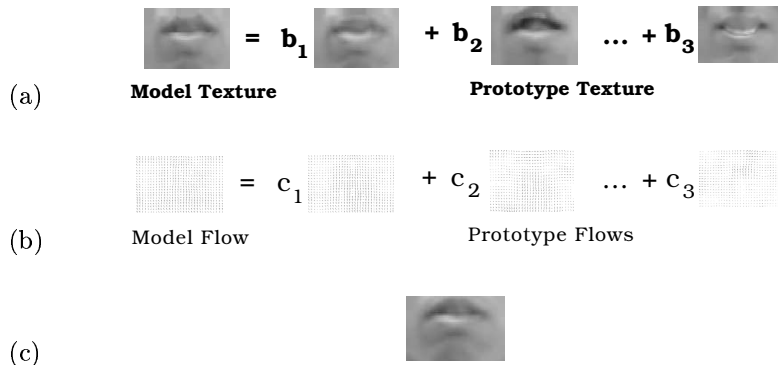
(a) $=\mathbf{b}_1$ ... $+\mathbf{b}_2$ ... $+\mathbf{b}_3$

Model Texture **Prototype Texture**

(b) $=c_1$ ... $+c_2$ ... $+c_3$

Model Flow Prototype Flows

(c)

Figure 3.2: A linear combination of images in the LMM framework involves (a) linearly combining the prototype textures using the coefficients **b** to yield a model texture and (b) the prototype flows using the coefficients **c** to yield a model flow. (c) The model image is obtained by warping the model texture along the model flow

3.1.2 Constructing an LMM for modeling mouths

In order to construct the single person mouth LMM we collected about 2066 images of mouths from one person. 93 of these images were manually chosen as example images to construct the LMM. The reference image can be chosen such that it corresponds to the average (in the vector space defined by the LMM) of the example images. However, the LMM can be defined only by choosing a reference. Therefore we take recourse to an iterative method where the reference image is initially chosen arbitrarily. Using this reference and the LMM that it defines, the average of the example images is computed. This average image then forms the reference image for the next step of the iteration. This method converges in a few iterations to a stable average image. A bootstrapping technique is used to improve the correspondence between the reference and other prototype images

Once the reference image is found and correspondence between the reference and prototypes established, we get a 93 dimensional LMM. The dimensionality of pixel space being 2066, the LMM constitutes a lower dimensional representation of the space (or manifold) of mouth images. However since many of the example images are alike there is likely to be a great deal of redundancy even in this representation. In order to remove this redundancy, we perform PCA on the example texture and shape vectors and retain only those principal components with the highest eigenvalues. As a result we obtain an LMM where a

novel texture is a linear combination of the principal component textures (which do not resemble any of the example textures) and similarly a novel flow is a linear combination of the principal component flows. A multiple person mouth LMM is obtained by following a similar procedure except we work with mouths of more than one person, in our case 3 different people of significantly different skin tone and mouth shape.

3.2 Learning to estimate the LMM parameters directly from images

The problem of estimating the matching LMM parameters directly from the image is modeled as learning a regression function from an appropriate input representation of the image to the set of LMM parameters. The input representation is chosen to be a sparse set of Haar wavelet coefficients while we use support vector regression as the learning algorithm.

3.2.1 Generating the Training Set

The training set was generated as follows.

- In the case of the single person mouth LMM each of the 2066 images is matched to the LMM using the SGD algorithm (for details of the SGD algorithm for matching LMMs to images see [27]). The two main parameters that need to be fixed at this step is the number of principal components of texture and flow space to retain and the number of iterations of the SGD algorithm. We found that retaining the top three principal components of the texture and flow space and allowing 250 iterations of SGD was sufficient to give us a good match and achieve minimum pixel error on average. Each image is thus represented as a six dimensional vector, which form the outputs for the learning problem.

- Each of the 2066 images is subject to the Haar wavelet transform and feature selection involving selection of those Haar coefficients with the highest variance as explained in section 2.3.2. We select 12 coefficients with the highest variance which form the inputs for the learning problem.

A similar procedure is followed for the multiple person mouth LMM, the only difference being that in this case we had to retain the top

eight principal components of the texture and flow spaces respectively, and require 1000 iterations of the SGD algorithm in order to get a reasonable reconstruction.

3.3 Results and Discussion

In this section we present the main results of our experiments. The results have two distinct aspects - the accuracy of the estimate and the quality of the image reconstruction. The accuracy of the estimate can be quantified and measured in two ways. 1) The closeness of the estimate from SVM regression to the one obtained from applying Stochastic Gradient Descent or 2) The error in the reconstructed image. The two are clearly related and yet one cannot be subsumed in the other.

The quality of the image reconstruction, on the other hand, is a phenomenological measure and although it is related to the both the above error measures, a great deal of work in vision and perception seems to indicate that image-based measures are not good indicators of perceptual likeness [44]. The phenomenological likeness of reconstruction is extremely important to the work on image synthesis. Since we are not directly concerned with image synthesis this measure has less significance for us. However, we will comment on both these aspects below.

3.3.1 Single Person Mouth Morphable Model

Mouth LMMs appear to work best when constructed for a single person. The results are distinctly better not only in terms of quantitative errors on pixels and LMM parameters but also in the quality of the reconstruction. In our experiments, we estimated six LMM coefficients corresponding to the top three principal components of the texture space and the flow space respectively. For each LMM coefficient a separate regression function had to be learnt.

Preliminary experiments indicated that Gaussian kernels were distinctly superior to estimating the LMM parameters in terms of number of support vectors and training error compared to polynomial kernels. As a result we next confined ourselves to experimenting with Gaussian kernels only. The free parameters in this problem, namely, the insensitivity factor ϵ, a weight on the cost for breaking the error bound C and the normalization factor σ were estimated independently for each of the LMM parameters using cross-validation. The regression function was used to estimate the LMM parameters of a test sequence of
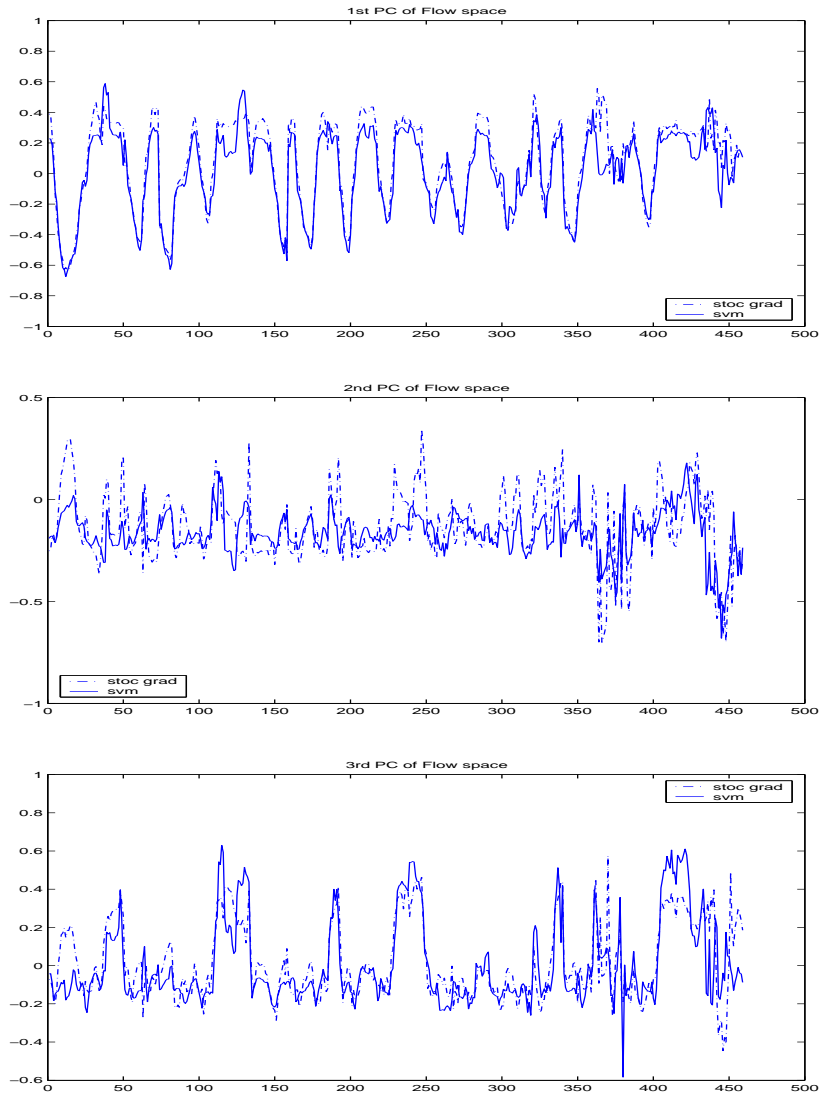
Figure 3.3: Estimates of a single person LMM flow parameters using the SGD algorithm and SVM regression on a test sequence of 459 images.
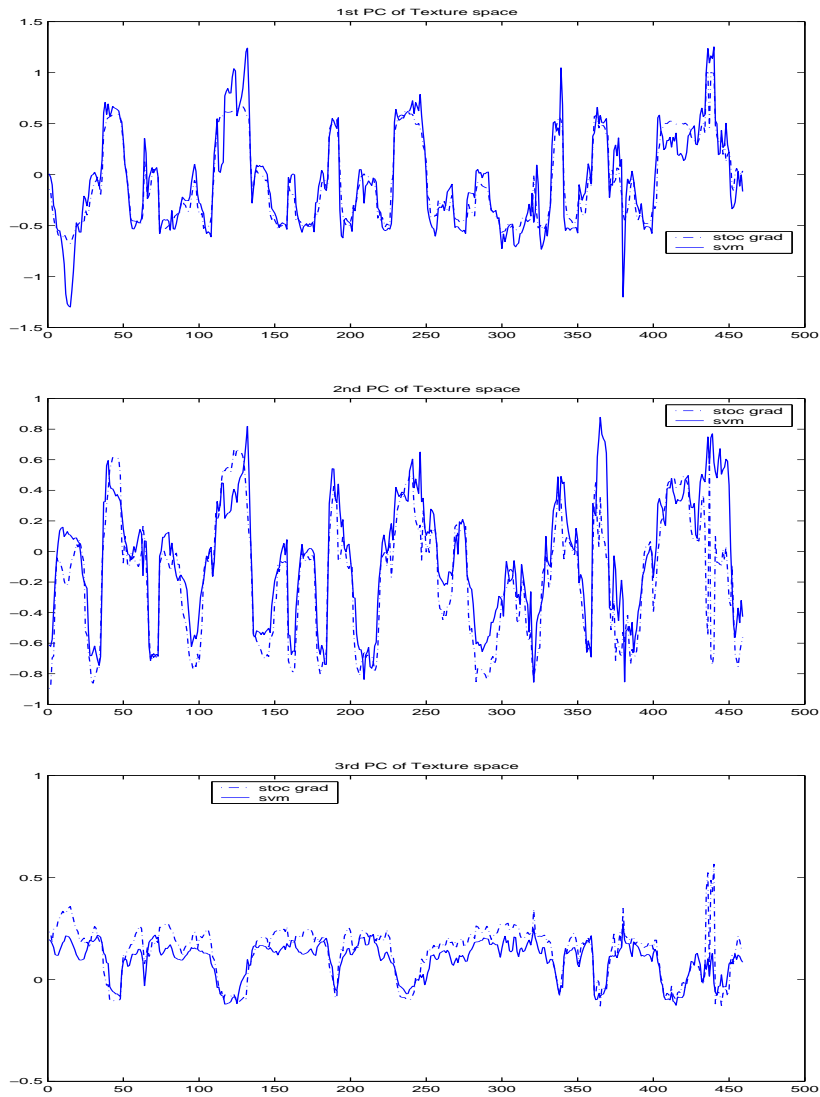
Figure 3.4: Estimates of a single person LMM texture parameters using the SGD algorithm and SVM regression on a test sequence of 459 images.

Figure 3.5: Estimates of (a) flow and (b) texture parameters of all the eight principal components of a multi-person LMM using the SGD algorithm and SVM regression on a test sequence of 104 images of individual 1.

Figure 3.6: Estimates of (a) flow and (b) texture parameters of all the eight principal components of a multi-person LMM using the SGD algorithm and SVM regression on a test sequence of 139 images of individual 2.

(a)   (b)
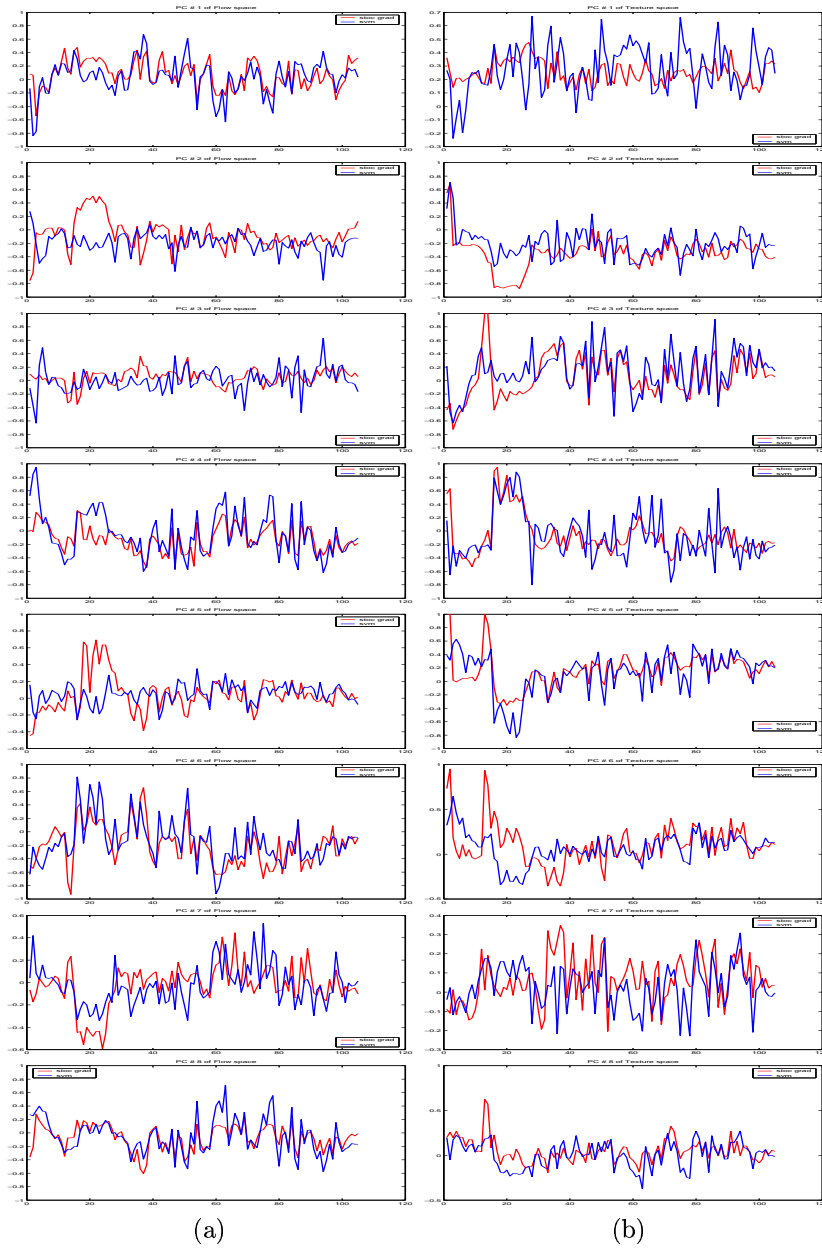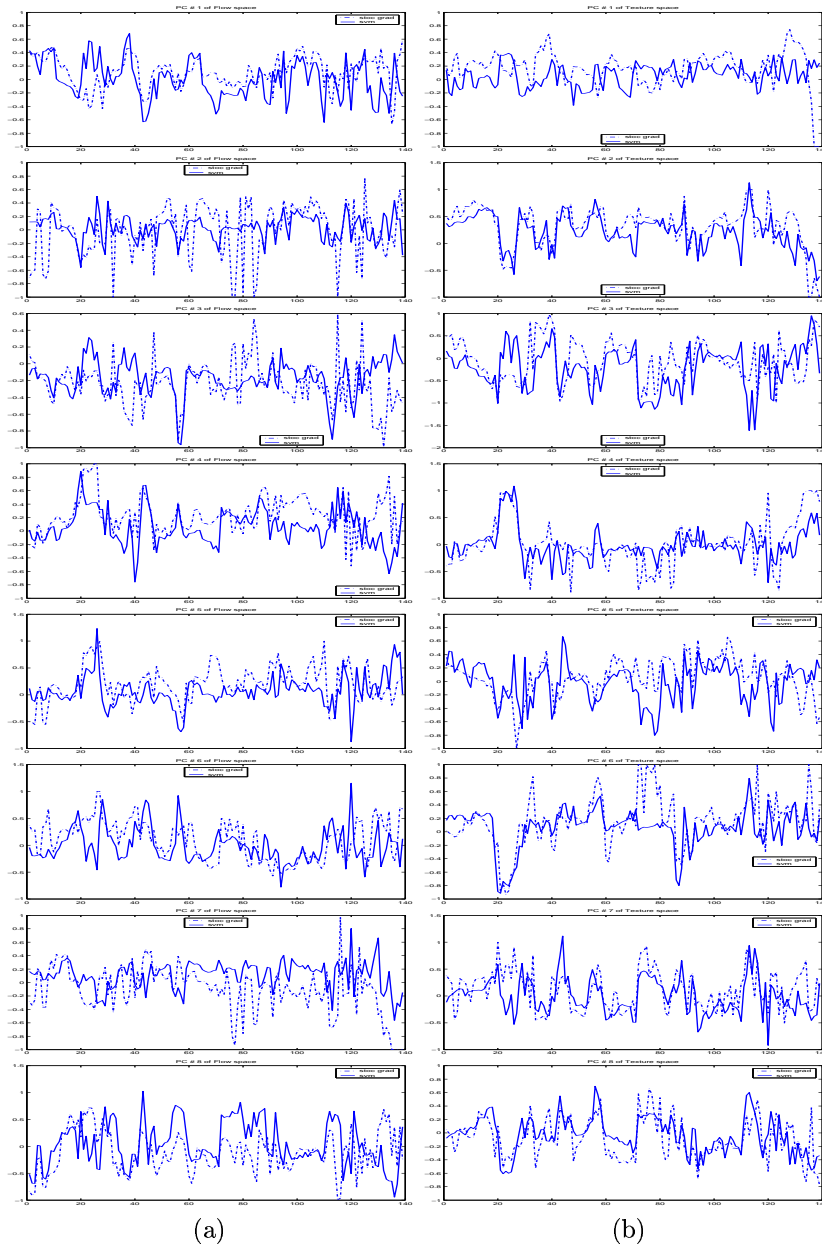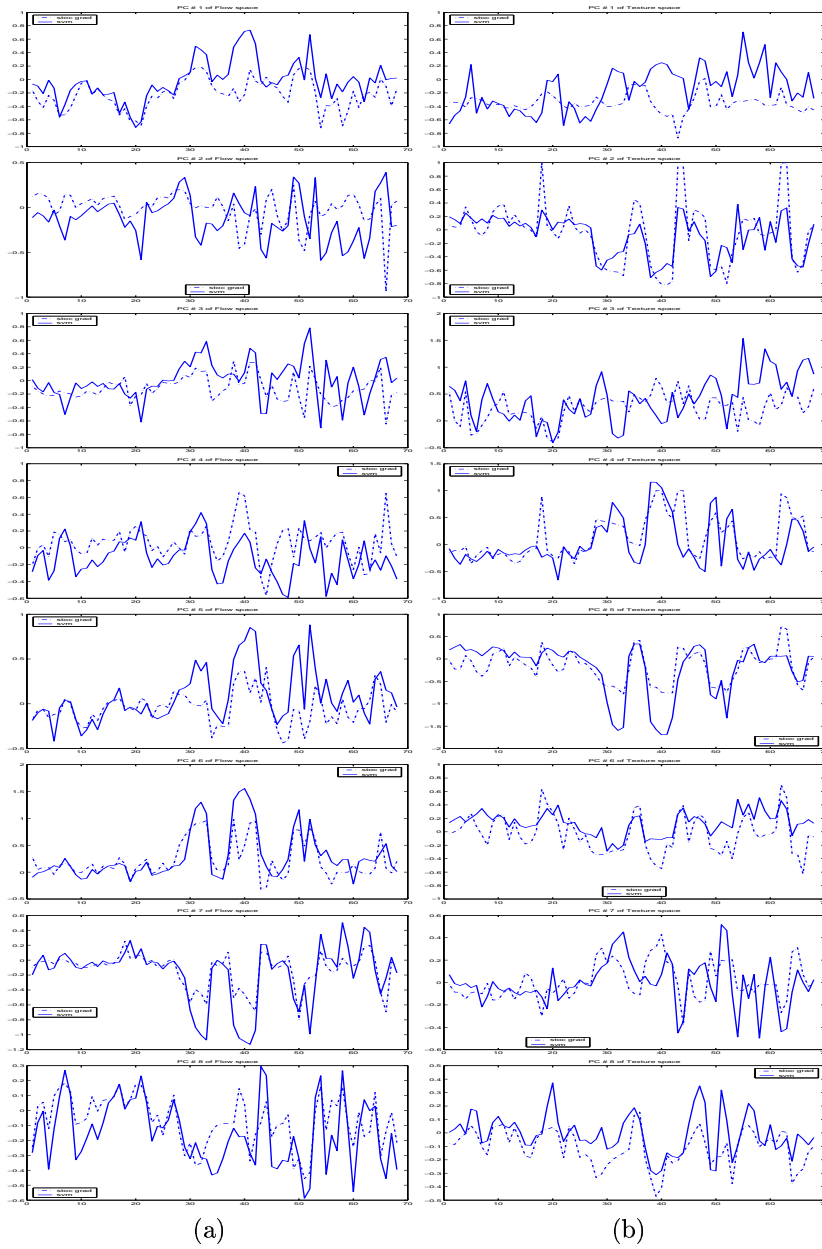
Figure 3.7: Estimates of (a) flow and (b) texture parameters of all the eight principal components of a multi-person LMM using the SGD algorithm and SVM regression on a test sequence of 69 images of individual 3.
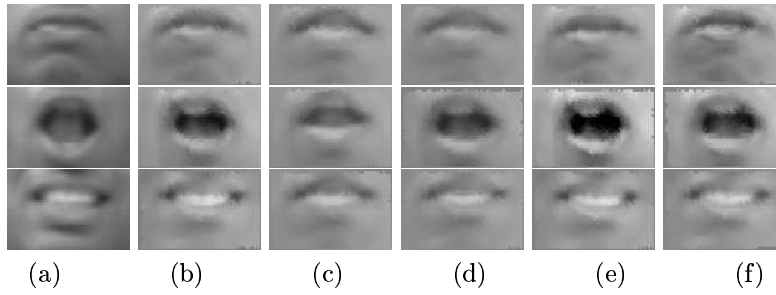
| (a) | (b) | (c) | (d) | (e) | (f) |

Figure 3.8: Comparison of reconstruction from LMM parameters estimated through SVM regression and through Stochastic Gradient Descent (SGD) for a single person mouth LMM. (a) Novel Image, (b) SVM Parameter Estimation followed by SGD for 10 iterations, (c) SGD for 10 iterations, (d) SGD for 50 iterations, (e) SVM Parameter Estimation alone and (f) the procedure which generates the training data i.e. SGD parameter estimation for 250 iterations.

| (a) | (b) | (c) | (d) | (e) | (f) |

Figure 3.9: Comparison of reconstruction from LMM parameters estimated through SVM regression and through Stochastic Gradient Descent (SGD) for a multiple (three) person mouth LMM. (a) Novel Image, (b) SVM Parameter Estimation followed by SGD for 100 iterations, (c) SGD for 100 iterations, (d) SGD for 500 iterations, (e) SVM Parameter Estimation alone and (f) the procedure which generates the training data i.e. SGD parameter estimation for 1000 iterations.
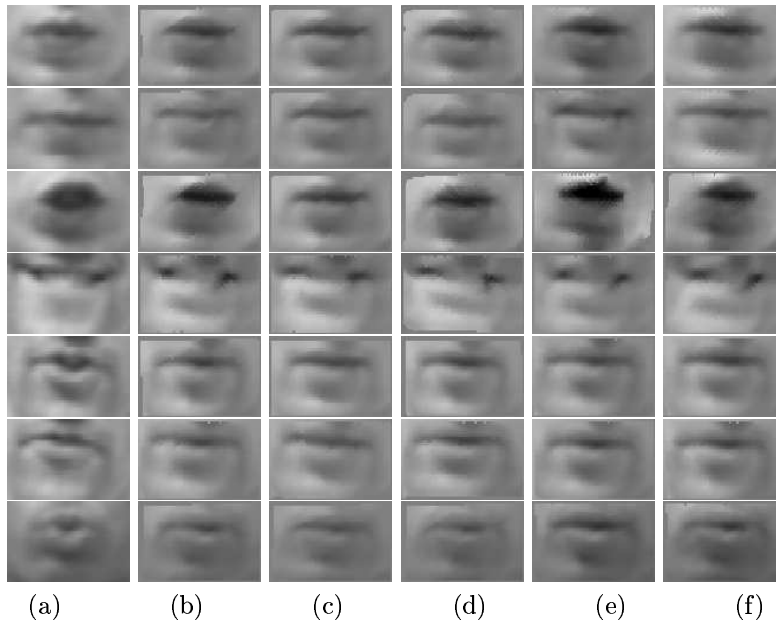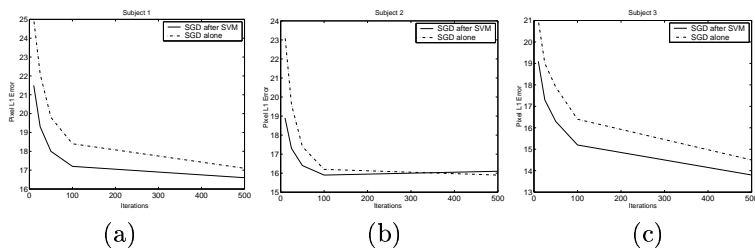
(a)   (b)   (c)

Figure 3.10: Plot of pixel error vs the iterations of the SGD algorithm. Comparing the case of initialization with the SVM regression estimate and with that of no initialization. The three plots correspond to the test sequences of the three individuals in the multiple person mouth LMM.

459 images. Figures 3.3 and 3.4 display the results for flow and texture parameters respectively where the performance of support vector regression is compared to that obtained using stochastic gradient descent. It shows the high degree of fidelity that one can get in estimating the LMM parameters directly from the image.

Examples of matches using the two methods shown in Figure 3.8 reveal two important aspects. 1) The speed advantage that one obtains by estimating the LMM parameters directly is quite clear. In only 10 iterations a SGD algorithm initialized with the estimate of an SVM regression is capable of achieving the same quality (and sometimes even better) than a SGD algorithm running solely for 50 iterations. 2) For those interested in synthesis issues, the quality of the reconstruction phenomenologically may leave a lot to be desired. However this is a quirk of trying to model mouths using the particular LMM described in Jones and Poggio [28]. Since this LMM involves morphing all prototypes to a single reference, it involves loss of information when dealing with objects such as mouths which have radically different views when open and closed. This results in poor quality when used for image synthesis. Fortunately, there are other morphable model which give better synthesis results (Ezzat and Poggio [41] and Cootes et al. [12]) and our method can in principle be extended to those.

3.3.2   Multiple Person Mouth Morphable Model

The case for the multiple person is roughly similar except that the results appear to degrade in quality. This is true of the capacity of the LMM to model the various mouth images as well as the SGD and

the SVM regression estimates. This is clear from the fact that we need to retain eight principal components of the texture and flow spaces, instead of just three in the case of the single person morphable model, to obtain reasonable reconstruction. Thus in our experiments we end up estimating 16 parameters in order to get reasonable reconstruction.

In Figure 3.9 we present examples of matches to novel images from test sets and compare the output of the SVM regression to that of the SGD algorithm. Comments similar to the single person case can be made in this case too. In Figure 3.10 we also present the speed up results in the form of average error over test set for a given number of iterations of the SGD algorithm in the two cases when initialized with the SVM regression estimate and when not.

With the growing application of morphable models in image analysis and synthesis, methods to estimate their parameters from images will become increasingly important. In this context, the learning-based approach to estimating the parameters of a LMM is natural as well as computationally feasible. In this chapter, we studied certain aspects of this method, namely the issue of single person vs multiple person, and speed up with respect to the SGD algorithm. However many other aspects of this problem need further study. So far our results indicate that the single person LMM is not only very good at modeling mouths, it is most amenable to direct estimation by learning. The case for the multiple person LMM is more ambiguous and it seems to under-perform the single person LMM in estimation accuracy as well as quality of reconstruction. One way to overcome to this problem may be to consider other models with better synthesis performance (Ezzat and Poggio [41] and Cootes et al. [12]). The claim that such estimation will aid in the design of real-time systems also needs to be tested.

Chapter 4

Applications to Man-Machine Interfaces

A new and interesting paradigm in Man-machine interaction is one of sociable humanoid robots, which can interact with human beings as social creatures using a variety of socially relevant modalities such as facial expressions, gestures, posture, gaze, voice, etc. This calls for, in the words of Cynthia Breazel [6], seamlessly combining insights from science, art and engineering. Principled ways of effecting such seamless combinations can form the design principles for building sociable robots.

One of the first steps in this direction would be to effect the design of systems that can combine a variety of constraints with a learning process to learn to understand (and respond) to important perceptual cues. In this thesis we have examined the possibility of such integration of top-down constraints with bottom-up learning by posing the problem as one of directly learning to estimate the parameters of morphable models from images. In this chapter, we shall discuss some of the applications of this technique, which seek to illustrate how they can be useful in the vision component of man-machine interaction.

4.1   Recognizing Facial Expressions

Ability to recognize, interpret and respond to facial expressions is considered to be a key component in man-machine interaction. Its role as a fundamental modulator of social interaction is least surprising given the seminal work of Ekman [16] which brings out the universal nature of fa-

45

cial expressions, in the sense that there are universal relations between particular facial configurations and certain emotions. These ideas have motivated the facial action coding system (FACS) [17] which posits facial action units as the "building blocks" of expressions. While this work is interesting, it has certain drawbacks. The main drawback being that the structure of FACS is not generative. Each facial action unit by itself appears to be ad-hoc in relation to the expressions/emotions that it is a part of. This can be interpreted as a lack of sufficient structure in FACS and therefore one can question the likelihood of estimating facial action units from facial configurations as an intermediary to expression recognition.

In the last chapter, we used a generative structure - the LMM - for modeling mouth patterns. We also showed that the parameters of a pixel LMM can be directly estimated from mouth images using learning. But this does not necessarily imply expression recognition. It is interesting to ask whether a generative structure such as an LMM is also capable of representing facial expressions in a natural way. We answer this question in the affirmative and as a consequence we propose the design of an expression recognition system that is capable of mitigating, at least in part, the familiar problem of manual annotation.

4.1.1 Expression Axes

The key idea behind the new expression recognition system is that of the *expression axes*. As explained in section 3.1.2 it is possible to take a corpus of mouth images and put them in correspondence within the structure of an LMM. It is also possible to perform PCA on the example textures and flows and express the LMM as a linear combination of a smaller set of the main principal components of the texture and flow spaces. Do these principal components represent something meaningful about mouth shapes? To answer this question, we morph the average mouth image along the three main flow principal components. The results are shown in Fig. 4.1. The interesting outcome of this exercise is that morphing along the first three principal components of flow space leads to recognizable mouth deformations such as open-close, smile-frown and pout-purse. These axes which we can call the expression axes can be used to ascertain the degree of expression of each kind in a novel mouth image. We can map each expression into a line-drawing LMM such as the one shown in Fig. 4.2. This line drawing LMM has 3 degrees of freedom corresponding to the 3 expression axes. Mapping the estimated parameters of the pixel LMM onto the line drawing LMM allows us to animate a cartoon figure to mimic the facial expressions of
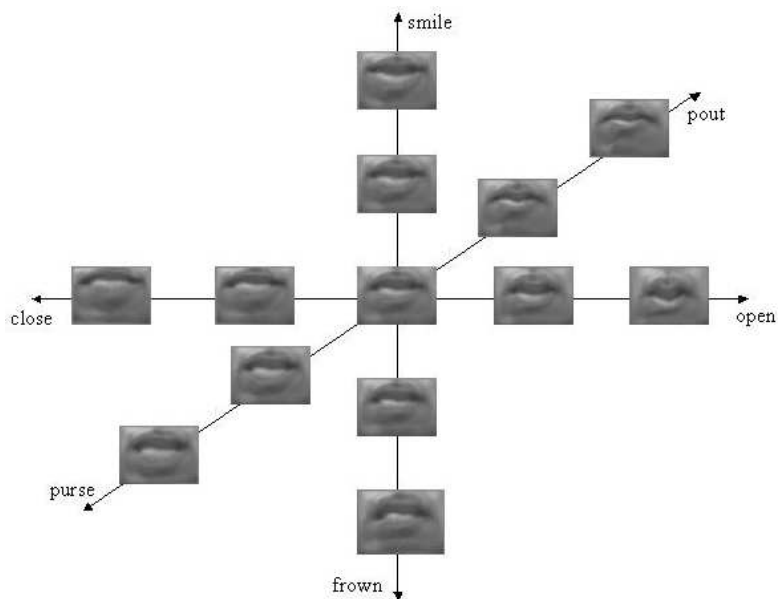
Figure 4.1: The Expression Axes showing the result of morphing the average mouth image along the first three flow principal components.

the person.

4.1.2 Experimental Details

We collected a corpus of 468 mouth images of a single person with varying degrees of expressions. 234 of these were used to construct a pixel LMM on the lines of section 3.1.2. PCA on the examples textures and flows allowed us to express the LMM in terms of a smaller number of texture and flow principal components. SGD was used to match the corpus of 468 images to the LMM. SVM regression was used to learn a mapping from a sparse wavelet-based representation (see section 2.3.2) to the expression axes (flow principal components) parameters using the Gaussian kernel, as detailed in section 3.3.1.

4.1.3 Results and Discussion

The resultant map was tested using a test set of 430 mouth images collected under a different circumstance (different day, time and lighting conditions) than the training set. The estimated expression parameters
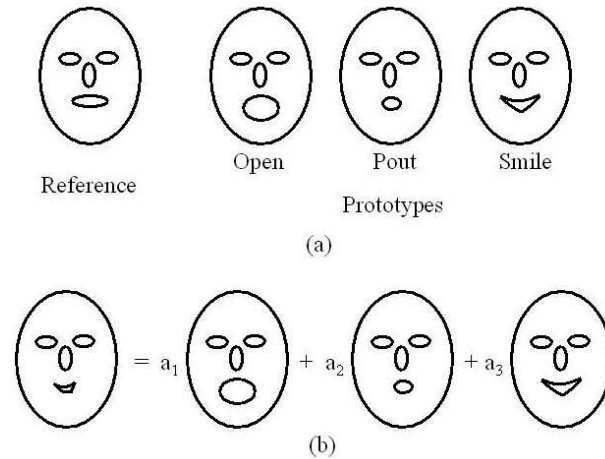
Figure 4.2: A line drawing LMM corresponding to the Expression Axes.

were mapped onto the line drawing LMM of Fig. 4.2 The result of this exercise is shown in Figs. 4.3 and 4.4.

The results are quite encouraging. They show the feasibility of extracting meaningful expressions from data without any significant human expert knowledge. The SVM regression training returns a small number of support vectors and therefore we can expect that real-time implementation will be easy. In principle, this method could be extended to more complex mouth shapes and also to other facial regions such as eyes. However some important questions could be raised and limitations pointed out.

An important question concerns the expressions represented by the expression axes. Do the deformations represented by the expression axes correspond to the ones that are perceived? Will they correspond to the expressions depicted by an artist? There is a straightforward way of dealing with these questions. Currently the map from the parameters of the pixel-LMM to the line drawing LMM of Fig. 4.2 is an identity map. Clearly that need not be the case. In general, we can conceive of the map from the expression axes to the line drawing LMM as a 3×3 matrix. This will require an artist to provide two renditions - 1) that of the facial expressions of openness, smile and pout, as line drawings and 2) line drawings that best approximate the expressions depicted by the expression axes. By expressing one set of parameters in terms of the other (using the SGD algorithm), we can compute the
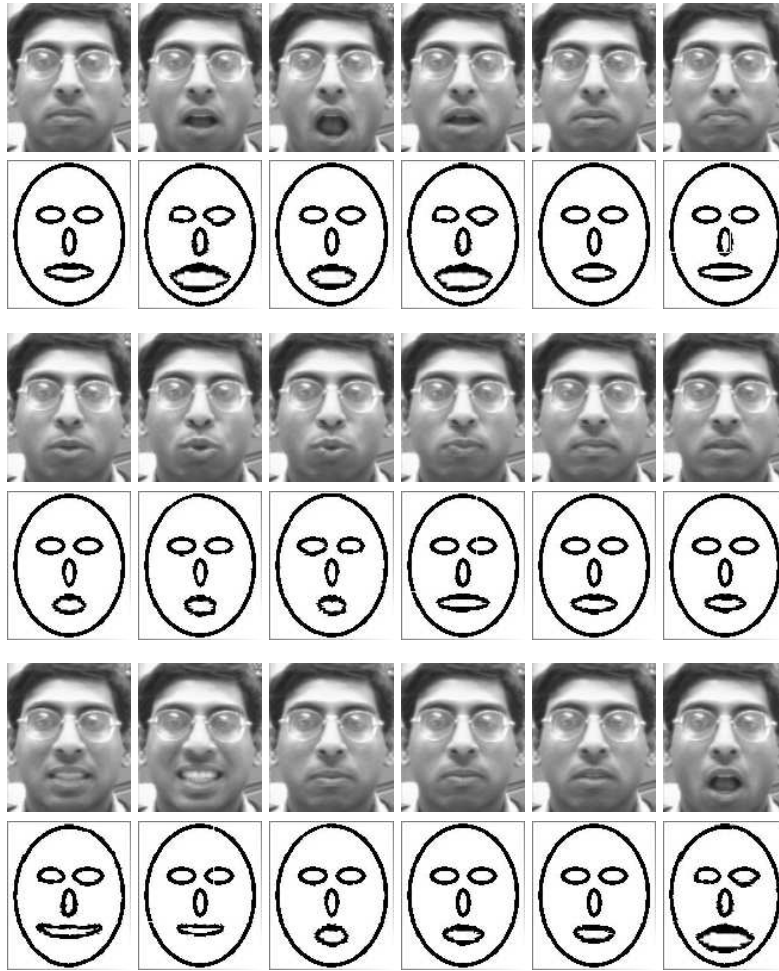
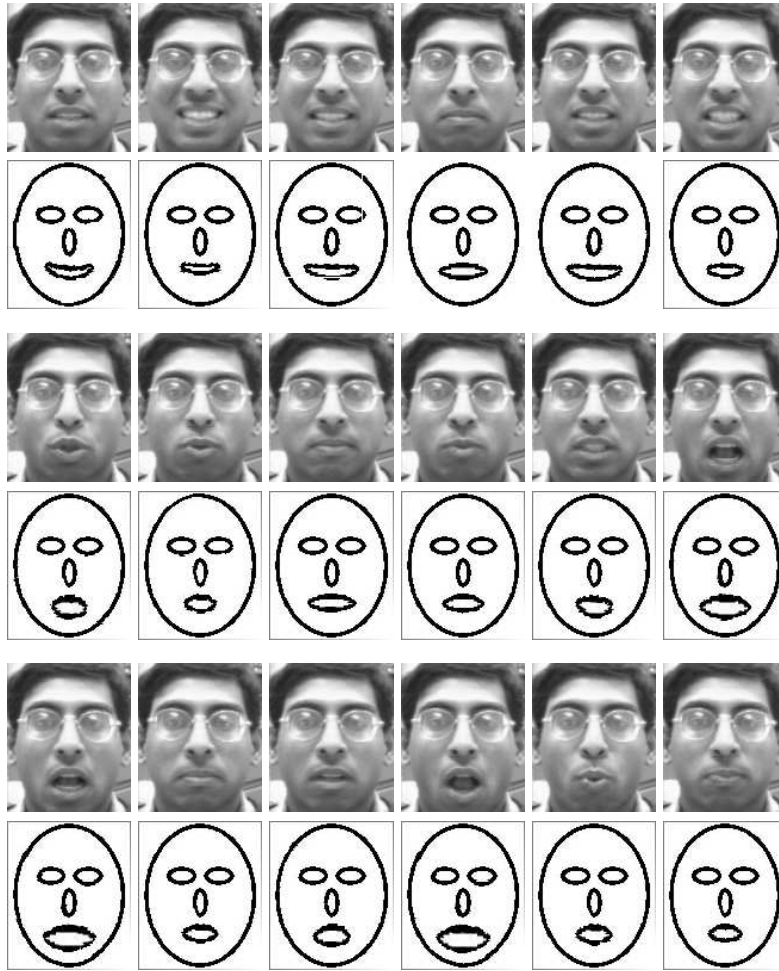Figure 4.3: Mapping the estimated expression axes parameters on a line drawing LMM.

Figure 4.4: Mapping the estimated expression axes parameters on a line drawing LMM.

matrix that transforms the expression axes parameters to the desired expression categories.

Another question is that of generalizability. We have seen in chapter 3 that the multiple person LMM performs less robustly than the single person. At the same time we have seen in chapter 2 that the learning based approach is able to generalize to more than one person. This leads to interesting questions. Should we construct multiple person LMMs in order to extract the expression axes? How representative is one persons expression axes to gauge the expressions of a different persons? These questions have much bearing on both the cognitive and engineering aspects of expression recognition systems.

4.2 Estimating Facial Pose

Facial Pose, along with gaze, is an important visual cue for the interpretation of human behavior and intentions. Pose estimation is a critical first step in one of the key tasks for man-machine interaction - pose independent face recognition. However, it is a very tricky task since the appearance of the face changes very significantly with the pose of the face. It is natural to assume that head pose being a three dimensional quantity its estimation from 2D images is inherently ambiguous. As a result pose estimation has been usually attempted by matching different kinds of models to the image. For example, in Beymer and Poggio [2] a pixel morphable model of faces capable of modeling pose variations is matched to novel face images. In case of Kruger et al. [30], an elastic graph is matched to Gabor wavelet-based facial features. Both of these rely on the analysis by synthesis algorithm to achieve the match of model to image.

Recent work by Sherrah et al. [25] has shown the possibility that there exists relationships between pixel patterns and the pose of a face. This work suggests that one should be able to learn a direct map from pixel-based representations to facial pose. In order, to make such a system self contained, we need to work with a model which is capable of modeling pose. In light of the application on expression recognition described in the last section, it is natural to expect that we model pose with the pixel LMM described in Beymer and Poggio [2]. However, such a model has several drawbacks. On the one hand, it is very difficult to construct since establishing correspondences between pixel images of faces in different pose is prone to errors. At the same time, it is fairly obvious that we do not need a complex pixel LMM to model pose. We suggest a much simpler, qualitative model based on a line drawing
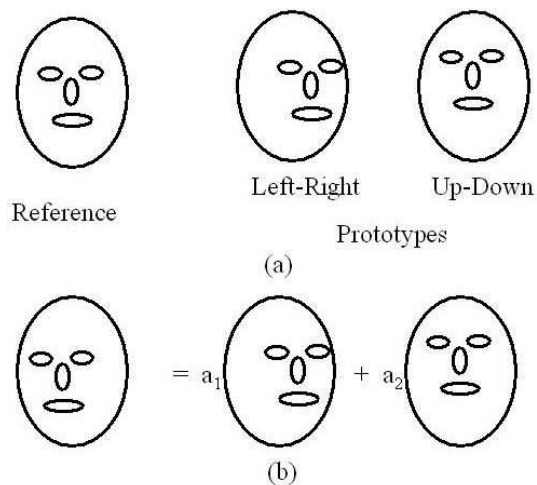
Figure 4.5: A line drawing LMM representing the pose space.

LMM. We expect that in most cases where pose information is needed, such a qualitative model is sufficient.

4.2.1 Representing Pose Using a Line Drawing LMM

A Line Drawing LMM is a special case of the LMM described in section 3.1. In a line drawing LMM, only the flow space exists, the texture space is a single point, i.e. there is just one texture which gets morphed to produce different shapes. A line drawing LMM which models the pose of a face is shown in Figure 4.5. The LMM has two degrees of freedom, one representing the left-right pose variation and the other the up-down pose variation. We shall call this LMM as the pose-LMM.

4.2.2 Matching Edge-maps to a Pose-LMM

How can we use the pose-LMM to estimate the pose of a face? On the face of it, it does not seem very likely since the pose-LMM is made up of simple line drawings whereas the face shows up as a richly textured image. But as shown by Jones and Poggio [28], it is possible to match the line drawing LMMs to novel and distorted line drawings. We extend this further and show that it is possible to match the pose-LMM to edge maps of face images. The matching algorithm is the same as the one used for matching the standard LMM except that since there are no

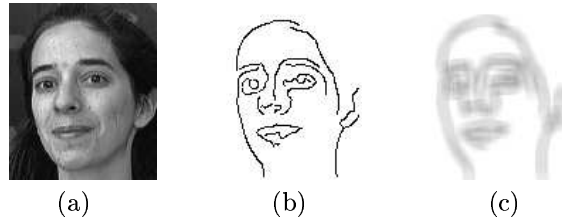<p style="text-align:center;">(a)   (b)   (c)</p>

Figure 4.6: (a) Face image, (b) Canny edge map and (c) Blurred edge map.

texture parameters to optimize over, the algorithm involves optimizing only over the flow and affine parameters (for details see [27]).

The face image is represented as a edge-map using a Canny edge-detector [10] and blurred using a Gaussian filter (Figure 4.6). The blurred image is matched to the pose-LMM using the SGD algorithm. The matching involves optimization over 8 parameters - 2 parameters representing the 2 degrees of freedom of the pose-LMM and 6 parameters for affine transformations. In Figure 4.7 we show the results of matching face images to the pose-LMM. As is obvious from the results the matching is not always accurate. This is mainly due to two reasons - 1) SGD algorithm gets trapped in local minima or 2) The edge detector output is either very noisy or misses crucial edges. In general we found that in about 25the cases, there is a visible mismatch between the pose of the face and that of the pose-LMM representation.

4.2.3 Learning the Parameters of the Pose-LMM

Since it is possible to estimate the pose in a majority of cases by matching the pose-LMM to a blurred edge-map of the face, the natural extension would be to ask if the relationship between pixel-based representation of faces and the pose-LMM can be learnt. We collected 356 face images from the FERET database [37] of different persons with various poses and in different and unknown lighting conditions. The blurred edge-map of these images was matched to the pose-LMM as explained in the last section. A section of this was considered for use as training examples. However we first prune the set by removing those images for which the SGD algorithm does not give acceptable results. This procedure gave us a training set of 233 images. The pixel-based representation is likely to involve pixels or wavelets. We experimented with coarse resolution pixel representations. We first subtract a best-fit brightness plane from the pixel values of the face images to correct
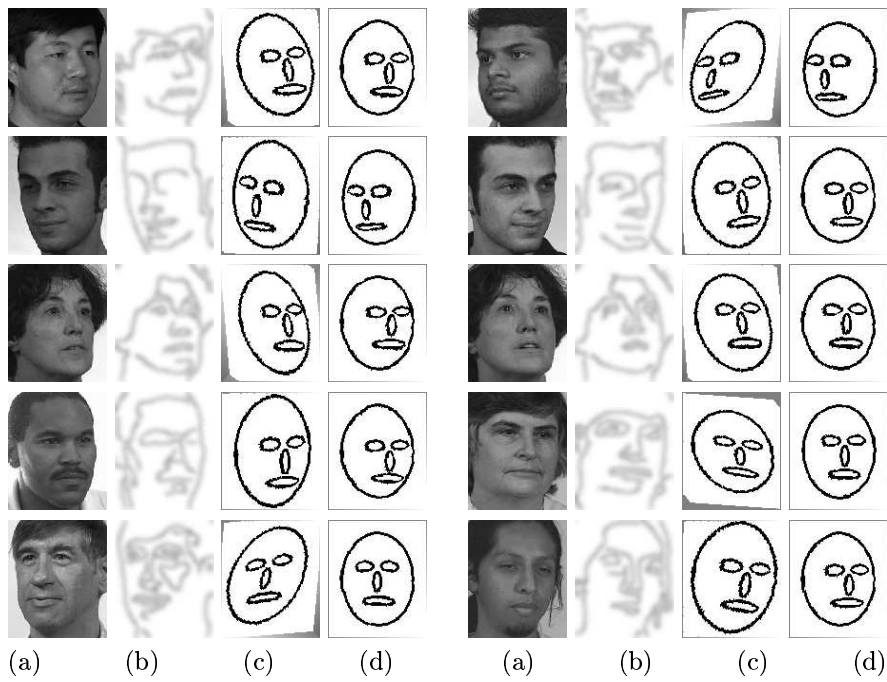
Figure 4.7: Results of matching of the pose-LMM to novel face images using SGD. (a) Face image, (b) Blurred Canny edge map, (c) Match obtained by the SGD algorithm and (d) Match after removing affine parameters.

for lighting variations. The face images of size 100 by 125 were then reduced to a size of 8 by 10. The resulting 80 coefficients formed the input for the learning problem.

4.2.4   Results and Discussion

The learning involves obtaining a map from the 80 dimensional input to the output i.e. the two pose dimensions. For learning, we used the familiar technique of SVM regression along with the Gaussian kernel. The free parameters of this problem were set by hand, it was found that the results are not particularly sensitive to the free parameters, over a wide range of the parameters. The resultant map was tested on a test set of 123 images, which included images of faces which were present in the training set, but in a different pose and also face images which were not present in the training set at all. In Figs. 4.8 and 4.9 we present examples of the results of testing on the two kinds of face images respectively.

The results in pose estimation so far are very encouraging. The most noteworthy part of this method is its simplicity at several levels. Both the top-down and bottom-up representations are simple and straightforward - the pose-LMM in case of the top-down and low resolution pixels in case of the bottom-up. Furthermore, we have managed to get these results with a surprisingly small number of examples. This method is capable of addressing the two main issues raised by the work of Sherrah et al. [25] namely that of a extensive database of labeled facial pose and that of their pose similarity space being discontinuous creating a lower bound on the pose resolution possible. Since the pose-LMM is a continuous space, the problem of a lower bound on the resolution is ruled out. The performance depends on the number of examples and the input representation.

Several possibilities need to be investigated before we can have a better picture of the strengths and weaknesses of this approach. In particular, the variability of this method with the number of examples and more importantly with the input representation. A wavelet based representation might turn out to be superior to low resolution pixels. Similarly, with increase in the number of examples the number of support vectors might increase which might make it difficult to apply this method for real-time applications.

| | (a) | (b) | (c) | | (a) | (b) | (c) |

Figure 4.8: Results of estimating the parameters of a pose-LMM directly from the image. These faces were also present in the training set but with different poses. (a) Face image, (b) Pose estimation using the SGD algorithm and (c) Pose estimation using SVM regression.
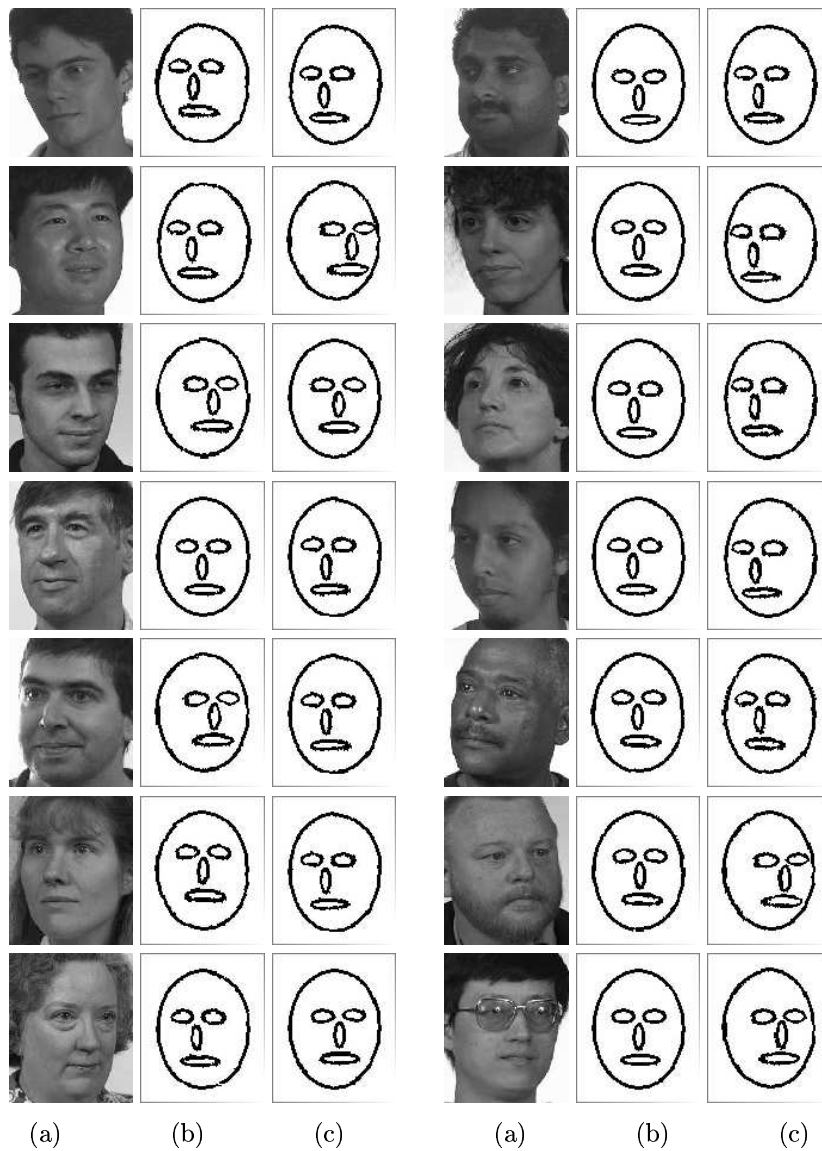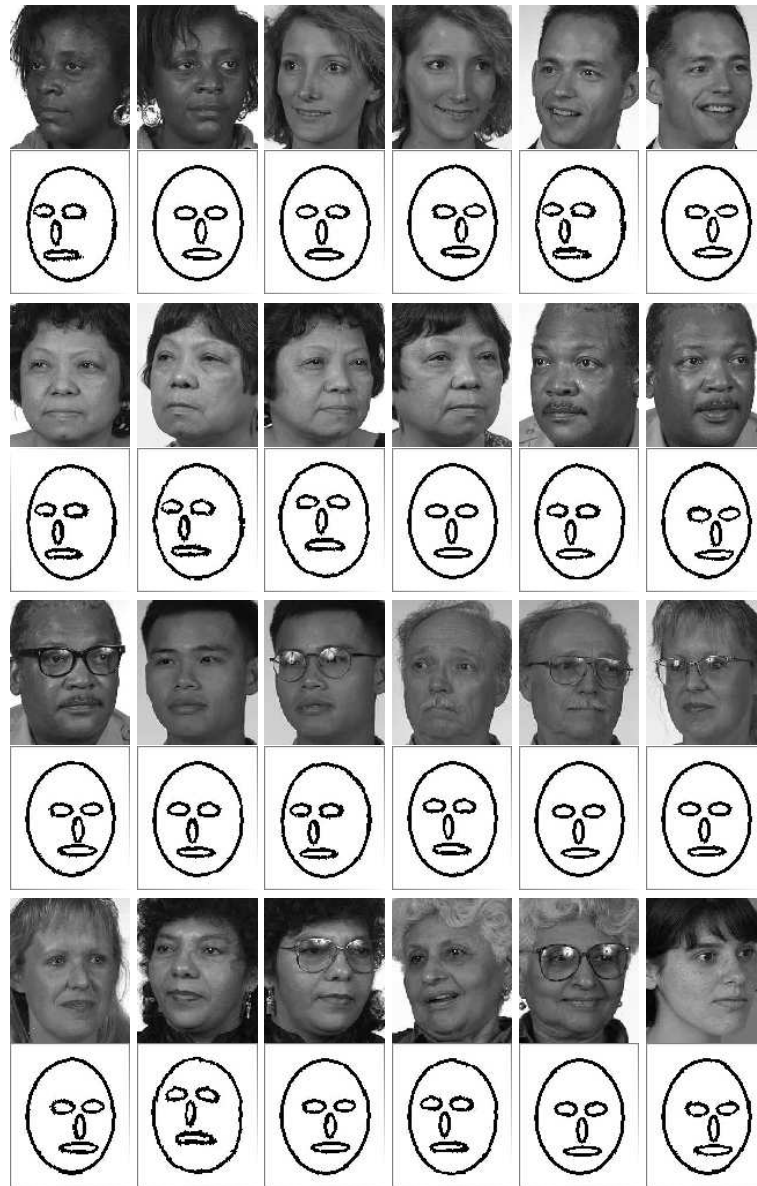
Figure 4.9: Results of estimating the parameters of a pose-LMM directly from the image. These faces were not present in the training set at all.
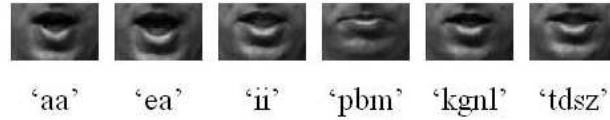
Figure 4.10: Images of visemes and their class.

4.3 Recognizing Visual Speech - Visemes

Recently Blanz [46] and Edwards et al. [21] have worked on using matching LMM parameters for higher level image analysis (or vision) applications such as face identification and shown encouraging results. In these applications the LMM parameter acts as an image feature that is then used to correctly classify the image. The claim therefore is that LMM parameters are better features than the normally assumed image features such as pixels or wavelets. (see section 1.3 and Figure 1.2). In order to test this claim, we introduce one such application, namely, viseme recognition.

Visemes are the visual analogues of phonemes (Ezzat and Poggio [41]). However, the mapping from phonemes to visemes is a many to one mapping. Different phonemes can lead to a single mouth shape and thus to a single viseme. Visemes like phonemes have temporal extent. However, so far we have investigated viseme recognition assuming visemes to be static images. Thus we consider the problem of mapping individual images to viseme classes.

4.3.1 Training Data

We used the visual speech corpus described in Ezzat and Poggio [41] for the viseme recognition problem. In this corpus, 39 phoneme classes which maps to 15 viseme classes have been identified. However, there was sufficient data to train for only six visemes classes (3 consonant classes and 3 vowel classes). Those six classes are 'pbm', 'tdsz', 'kgnl', 'ii', 'ea', 'aao' and a prototypical image of each class is shown in Figure 4.10. Details about these classes and their sizes are given are given in Table 4.1. The training and testing sets were obtained by dividing these sets roughly in the ratio of 2:1.

Viseme class	Associated phonemes	Size (Number of examples)
pbm	p, b, m	106
tdsz	t, d, s, z, th, dh	170
aao	aa, o	70
ii	ii, i	65
ea	e, a	54
kgnl	k, g, n, l, ng, y	124

Table 4.1: The viseme classes and their associated phonemes and the size of the training and testing sets.

4.3.2   Implementation Details

Two different feature representations were investigated as input for classification, namely, wavelets and LMM parameters. The two representations lead to somewhat different approaches for viseme recognition and the idea is to compare the relative merits of these two approaches and therefore the two representations.

- **Case 1: Haar Wavelet Representation.** This method utilizes an image-based representation in which a linear (multi-class) SVM classifier was trained to accept wavelet coefficients of the mouth image as input and to output the viseme class. The inputs were determined by selecting twelve low resolution Haar wavelet coefficients using the method described in section 3.2.1.

- **Case 2: LMM Representation.** This method relies on a LMM as an intermediate representation. For this purpose, 91 images from the corpus were used to construct a single person mouth LMM and the model was matched to the remaining images using the SGD algorithm. The top principal components of the flow and texture spaces were now used as a feature set to represent each image. A linear (multi-class) SVM classifier was trained to accept the matching LMM parameters of the mouth image as input and to output the viseme class.

Since this is a multi-class problem, we have experimented with the top-down decision graph (Fig. 4.11, see Nakajima, et al. [8]) as the multi-class strategy. This strategy involves the training of a classifier to distinguish between any two visemes, each of which is a linear SVM. We have compared the performance of this technique with the k-nearest neighbors technique.
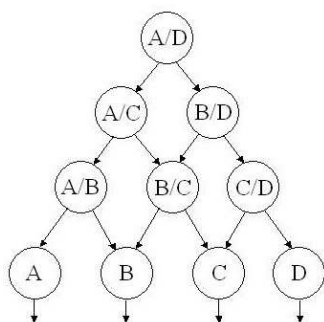
Figure 4.11: Graphical representation of the top-down multi-class classification strategy. Each non-leaf node represents a classifier.

4.3.3 Results and Discussion

The results of viseme recognition vary with the dimension of the input representation. We kept the input dimension for the wavelet representation constant at 12 and varied that of the LMM representation. We found that as long as the input dimension was less than 12, the LMM representation under-performed compared to the wavelet representation and gave the same result when the input dimension was brought to 12. This probably means that the representational power of the LMM is almost identical to that of wavelets. The results comparing the different representations and different multi-class strategies for an input dimension of 12 are presented in Tables 4.2 and 4.3. Since the data sets are quite small, the results presented are averages over 100 random 2:1 splits of the examples.

The idea of using the morphable model parameters for higher level vision and interpretation is gaining ground. However, our results with viseme recognition seem to seem to cast a doubt on the utility of this approach. Our previous examples of expression recognition and pose estimation seem to indicate the utility of using the LMM parameters as the higher-level parameters to be estimated. But attempting to use these parameters as features for estimating some other higher-level categories does not seem to buy anything substantial. We conjecture that *LMM parameters cannot be a substitute for image features*. However this could merely be a peculiarity of this particular application or our data set, and so much more work needs to be done to ascertain the above conjecture.

	pbm	tdsz	aao	ii	ea	kgnl
pbm	0.94	0.03	0.0	0.01	0.0	0.02
tdsz	0.01	0.82	0.0	0.04	0.0	0.14
aao	0.01	0.04	0.77	0.04	0.11	0.03
ii	0.04	0.15	0.03	0.58	0.03	0.15
ea	0.02	0.13	0.26	0.10	0.36	0.14
kgnl	0.03	0.41	0.05	0.11	0.07	0.33

(a)

	pbm	tdsz	aao	ii	ea	kgnl
pbm	0.95	0.02	0.00	0.02	0.00	0.01
tdsz	0.00	0.86	0.00	0.02	0.00	0.12
aao	0.00	0.00	0.83	0.01	0.11	0.05
ii	0.04	0.08	0.00	0.67	0.02	0.19
ea	0.00	0.10	0.07	0.08	0.57	0.18
kgnl	0.01	0.38	0.03	0.09	0.07	0.43

(b)

	pbm	tdsz	aao	ii	ea	kgnl
pbm	0.91	0.05	0.0	0.0	0.01	0.02
tdsz	0.01	0.90	0.0	0.02	0.0	0.08
aao	0.0	0.04	0.77	0.02	0.16	0.05
ii	0.01	0.12	0.02	0.62	0.05	0.11
ea	0.03	0.07	0.31	0.16	0.26	0.16
kgnl	0.02	0.40	0.03	0.11	0.08	0.36

(c)

	pbm	tdsz	aao	ii	ea	kgnl
pbm	0.95	0.03	0.00	0.00	0.01	0.01
tdsz	0.00	0.89	0.00	0.01	0.01	0.08
aao	0.00	0.00	0.81	0.02	0.13	0.04
ii	0.03	0.10	0.01	0.66	0.07	0.13
ea	0.01	0.08	0.23	0.11	0.42	0.15
kgnl	0.00	0.36	0.03	0.07	0.08	0.46

(d)

Table 4.2: Confusion matrices for (a) LMM-representation, k nearest neighbor, k = 4, (b) LMM-representation, linear SVM, top-down multi-class, (c) wavelet-representation, k nearest neighbor, k = 4 and (d) wavelet-representation, linear SVM, top-down multi-class.

	Linear SVM Top-Down	k nearest neighbors			
		k = 1	k = 2	k = 3	k = 4
LMM Representation	0.73	0.64	0.65	0.66	0.68
Wavelet Representation	0.73	0.65	0.66	0.67	0.68

Table 4.3: Overall accuracy of viseme recognition.

Chapter 5

Conclusions and Future Work

The paradigm of *embodied intelligence* advanced by many researchers, but notably by Rodney Brooks and co-workers [39] seeks to understand human intelligence and behavior as the result of four intertwined attributes, namely, developmental organization, social interaction, embodiment and physical coupling, and multi-modal integration. In the framework of embodied intelligence, intelligent systems is sought to be built by having the system directly (and physically) coupled with its environment and allowing the system to develop in ways that allow for the coupling to be exploited in ways beneficial to the system. This framework departs from the framework of classical AI which seeks to build intelligent systems as some kind of symbol processors.

5.1 The Big Picture

A natural question within the framework of embodied intelligence is how can the coupling between system and environment be achieved. There are two levels at which this question could be asked. At the level of embodiment, the system must be capable of representing the environment and at the level of development, the system must be able to learn the various representations involved and the ways of mapping between them. While these ideas have led to significant advances in the realm of humanoid robots and related sensori-motor tasks, their relevance for perception itself has not been fully explored. We believe that our work is a small step in clarifying how systems can be embodied
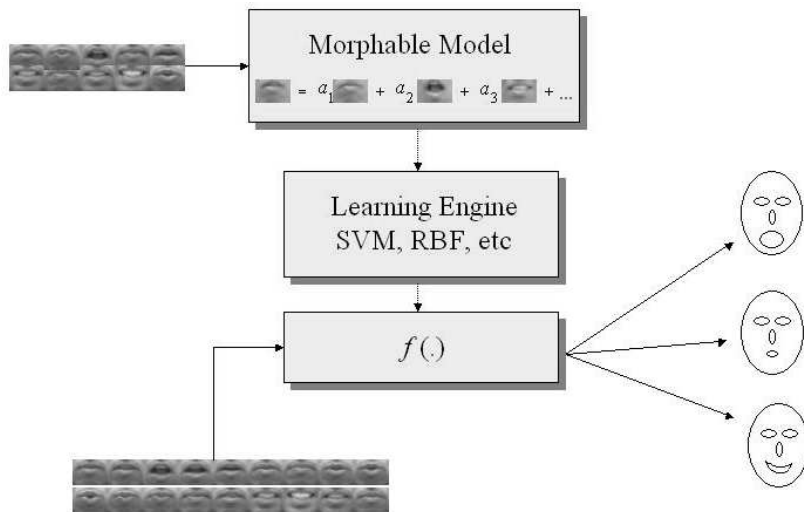
Figure 5.1: Combining top-down constraints with bottom-up learning for perception.

and developed for perceptual tasks.

We have suggested that one way of achieving a coupling between environment and perceiver is to be able to represent the categories of interest in the environment (which can be understood as the top-down knowledge in an organism) and use learning to map from retinal information to the parameters (which is understood as the bottom-up process) within the top-down representation. We illustrate this big picture in Fig. 5.1.

5.2   Possible New Systems

There are several directions that future research might take. We list a few possibilities below.

- Expand the scope of this method by implementing similar systems for other perceptual tasks, e.g. tracking gaze and eye blinks.

- Investigate the relative efficacies of pixel LMMs and line drawing LMMs to model images. So far we have used the pixel LMM for expression recognition and the line drawing LMM for pose estimation. Nothing prevents us from using the pixel LMM for pose and a line drawing LMM for expressions.

- Build a virtual actor system which is able to analyze a persons expressions and map them onto a system for synthesizing realistic images of an actor (Ezzat and Poggio [41]).

- Extend this method to other objects of interest such as cars, indoor objects, etc.

5.2.1 Related Scientific and Philosophical Questions

This work throws up many questions for computational cognitive science. The definitive links to embodied intelligence can be drawn if these ideas can be integrated with the kind of sensori-motor capabilities that humanoid robots have. For example, can these ideas be used to map scene features to action parameters? Do we need to go through internal representations? If so, are these representations natural and embodied?

Bibliography

[1] P. Hallinan A. Yuille and D. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.

[2] D. Beymer and T. Poggio. Face recognition from one example view. In *Proceedings of the International Conference of Computer Vision*, pages 500–507, Cambridge, MA, June 1995.

[3] D. Beymer and T. Poggio. Image representations for visual learning. *Science*, 272(5270):1905–1909, June 1996.

[4] D. Beymer, A. Shashua, and T. Poggio. Example based image analysis and synthesis. A.I. Memo No. 1431, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1993.

[5] M.J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 1996. Revised preprint.

[6] C. Breazel. *Designing Sociable Robots*. MIT Press, Cambridge, 2002.

[7] J.S. Bruner. On perceptual readiness. *Psychological Review*, 64:123–152, 1957.

[8] C. Nakajima, M. Pontil, B. Heisele and T. Poggio. People Recognition in Image Sequences by Supervised Learning. *MIT AI Memo No. 1688/CBCL Memo No. 188*, 2000.

[9] C. Papageorgiou, M. Oren and T. Poggio. A general framework for object detection. In *Proceedings of the International Conference on Computer Vision*, Bombay, India, January 1998.

[10] F. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[11] P. Cavanagh. What's up in top-down processing. In *Representations of Vision*. Cambridge Univ. Press, 1991.

[12] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, Freiburg, Germany, 1998.

[13] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:1–25, 1995.

[14] R. Freund E. Osuna and F. Girosi. Training support vector machines: an application to face detection. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

[15] K. Ebihara, J. Ohya, and F. Kishino. Real-time facial expression detection based on frequency domain transform. *Visual Communications and Image Processing, SPIE*, 2727:916–925, 1996.

[16] P. Ekman. Facial expressions. In *Handbook of Cognition and Emotion*, pages 301–320, Sussex, UK, 1999.

[17] P. Ekman and E.L. Rosenberg (Eds.). *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS)*. Oxford University Press, New York, 1997.

[18] Irfan A. Essa and Alex Pentland. A vision system for observing and extracting facial action parameters. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 76–83, Seattle, WA, 1994.

[19] Tony Ezzat. Example-based analysis and synthesis for images of human faces. Master's thesis, Massachusetts Institute of Technology, 1996.

[20] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–269, 1995.

[21] C.J. Taylor G.J. Edwards and T.F. Cootes. Learning to identify and track faces in image sequences. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 260–265, 1998.

[22] H.P. Graf, T. Chen, E. Petajan, and E. Cosatto. Locating faces and facial parts. *Proc. Int. Workshop on Automatic Face- and Gesture-Recognition*, pages 41–46, 1995. In M. Bichsel (editor).

[23] S. Baluja H.A. Rowley and T. Kanade. Neural network based face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20:23–38, 1998.

[24] M. Isard and A. Blake. Contour-tracking by stochastic propagation of conditional density. In *Proceedings of European Conference on Computer Vision*, pages 343–356, 1996.

[25] S. Gong J. Sherrah and E. Ong. Understanding pose discrimination in similarity space. In *10 th British Machine Vison Conference*, volume 2, pages 523–532, Nottingham, UK, 1999.

[26] C. Jacobs, A. Finkelstein, and D. Salesin. Fast multiresolution image querying. In *SIGGRAPH '95 Proceedings*, 1995. University of Washington, TR-95-01-06.

[27] M. Jones and T. Poggio. Model-based matching by linear combinations of prototypes. A.i. memo, MIT Artificial Intelligence Lab., Cambridge, MA, 1996.

[28] M. Jones and T. Poggio. Multidimensional morphable models: A framework for representing and matching object classes. In *Proceedings of the International Conference on Computer Vision*, Bombay, India, 1998.

[29] W. Kohler. *Gestalt Psychology*. Liveright Publishing, New York, 1947.

[30] N. Kruger, M. Potzsch, and C. von der Malsburg. Determination of face position and pose with a learned representation based on labeled graphs. *Image and Vision Computing*, 15(8):665–673, August 1997.

[31] H.C. Lee and R. M. Goodwin. Colors as seen by humans and machines. In *ICPS: The Physics and Chemistry of Imaging Systems*, pages 401–405, May 1994.

[32] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[33] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman and Company, San Francisco, 1982.

[34] T. Vetter M.J. Jones, P. Sinha and T. Poggio. Top-down learning of low-level vision tasks. *Current Biology*, 7(11):991–994, 1997.

[35] D. Mumford. Pattern theory: a unifying perspective. In *Perception as Bayesian Inference*. Cambridge Univ. Press, 1996.

[36] N. Oliver, F. Berard, and A. Pentland. Lafter: Lips and face tracker. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 123–129, Puerto Rico, 1996.

[37] J. Huang P.J. Phillips, H. Wechsler and P. Rauss. The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.

[38] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

[39] I. Robert C. Kemp M. Marjanovic B. Scassellati R. Brooks, C. Breazel (Ferrell) and M. Williamson. Alternative essences of intelligence. In *Proceedings of AAAI98*, pages 961–967, Madison, WI, 1998.

[40] K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20:39–51, 1998.

[41] T. Ezzat and T. Poggio. Visual Speech Synthesis by Morphing Visemes. *MIT AI Memo No. 1658/CBCL Memo No. 173*, 1999.

[42] Demetri Terzopoulos and Keith Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):569–579, 1993.

[43] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems*. W. H. Winston, Washington, D.C., 1977.

[44] S. Ullman. *High-Level Vision*. MIT Press, Cambridge, MA, 1996.

[45] S. Ullman and R. Basri. Recognition by linear combination of models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:992–1006, 1991.

[46] V. Blanz. *Automated Reconstruction of Three-Dimensional Shape of Faces from a Single Image*. Ph.D. Thesis (in German), University of Tuebingen, 2000.

[47] V. Kumar and T. Poggio. Learning-Based Approach to Real Time Tracking and Analysis of Faces. In *Proceedings of the Fourth International Conference on Automatic Face and Gesture Recognition*, pages 96–101, Grenoble, France, 2000.

[48] S.E. Golowich V. Vapnik and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. In *Advances in Neural Information Processing Systems*, volume 9, pages 281–287, San Mateo, CA, 1997. Morgan Kaufmann Publishers.

[49] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[50] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, Berlin, 1982.

[51] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[52] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:7:733–742, 1997.