MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY
and
CENTER FOR BIOLOGICAL AND COMPUTATIONAL LEARNING
DEPARTMENT OF BRAIN AND COGNITIVE SCIENCES

A.I. Memo No. 1672 September, 1999
C.B.C.L Paper No. 179

Learning-Based Approach to Real Time Tracking and Analysis of Faces

Vinay P. Kumar and Tomaso Poggio

This paper 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 invariance properties of image representations based on Haar wavelets to robustly capture various facial features. Similarly, unlike previous approaches this system 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.

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.

This report describes research done within the Center for Biological and Computational Learning in the Department of Brain and Cognitive Sciences and at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. This research is sponsored by a grant from the National Science Foundation under contract No. IIS-9800032. Additional support is provided by Eastman Kodak Company, Daimler-Benz, Siemens Corporate Research, Inc. and AT&T.

1 Introduction-Motivation

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. It is a challenging problem 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.

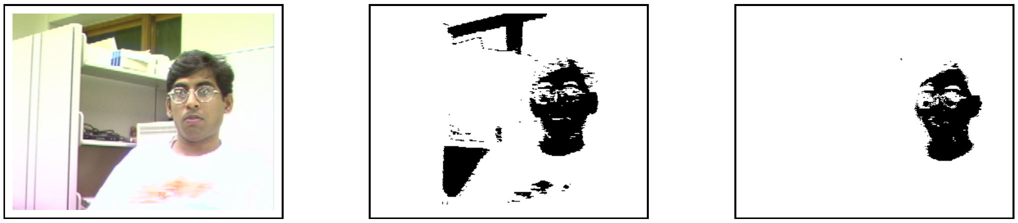
1. Detect and localize faces in a cluttered background.
2. Detect and localize different facial features such as eyes, nostrils and mouth.
3. Analyze these regions to estimate suitable parameters.

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 (Essa and Pentland [10], Terzopoulos and Waters [18], Yuille et al. [1]) and motion estimation (Black and Yacoob [4], Ezzat [11]). 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. [3], Beymer and Poggio [2] and Cootes et al. [6]). However these view-based models are explicitly dependent on correspondence between facial features and therefore repeated estimation of this correspondence during analysis. In this paper, we describe 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. [13] and pedestrian detection in Papageorgiou et al. [5]. 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



(a) (b) (c)

Figure 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.

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. [12] and Ebihara et al. [9] 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. [8]. 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 [7]) 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. SVM classification is also used to detect eyes. The problem of learning the regression function from the sparse subset of wavelet coefficients of the mouth pattern to the desired output is tackled by SVM regression.

The present 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. The system performs at approximately 10 frames/second on a dual-Pentium processor machine.

2 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 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 [14]) 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 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.

2.2 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. [8].

- 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.

3 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

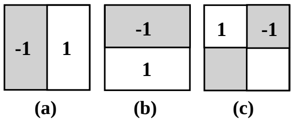


Figure 2: The 3 types of 2 dimensional non-standard Haar wavelets; (a) "vertical", (b) "horizontal" and (c) "corner".

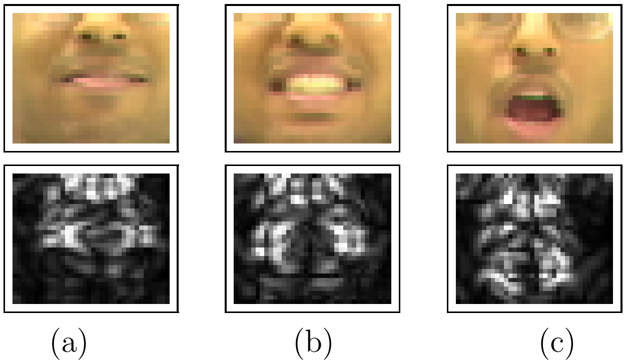


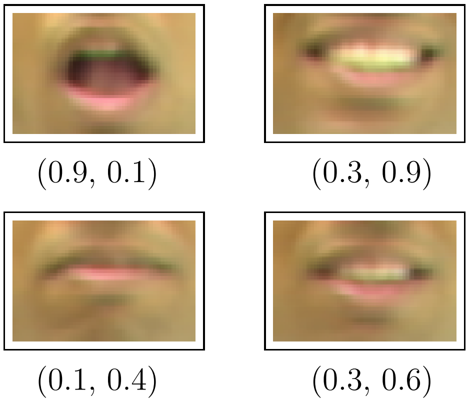
Figure 3: Illustrating the corner type Haar wavelet response to 3 generic mouth shapes. Note that the nostrils and the contours of the mouth stand out.

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. [16]) or on optical flow (Black and Yacoob [4]) 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.

3.1 Encoding localized variations using Haar wavelets

Wavelets can be interpreted as encoding image variations locally in space and spatial frequency. In figure 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 [15]) 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



(0.9, 0.1) (0.3, 0.9)

(0.1, 0.4) (0.3, 0.6)

Figure 4: Illustrating subjective annotation of the training set for degrees of (openness, smile).

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 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.
2. As features for analysis of mouth region.

3.2 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

$$\mathbf{x}^* = \arg\max_k P(\mathbf{x}_k | y(\mathbf{x}_k) > 0) \tag{1}$$

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 [17]),

$$P(\mathbf{x}|y(\mathbf{x}) > 0) \approx P(y(\mathbf{x})|y(\mathbf{x}) > 0) \approx P(y(\mathbf{x})|\mathbf{x} \in S_p) \tag{2}$$

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.

4 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.

4.1 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. [3] and Beymer and Poggio [2]). This is done based on the subjective interpretation of the user and no actual image parameters are measured. Figure 4 shows some examples of such a subjective annotation.

4.2 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 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 1: Normalized variance values for the Haar coefficients of mouth sequences (as they open and close). (a) Vertical, (b) Horizontal and (c) Corner types.

4.3 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,

$$\mathbf{V} = \begin{bmatrix} {\mathbf{v}^{(1)}}^T \\ {\mathbf{v}^{(2)}}^T \\ \vdots \\ {\mathbf{v}^{(N)}}^T \end{bmatrix} \tag{3}$$

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.

4.4 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. [19] and Vapnik [20]) 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:

$$f(\mathbf{x}, \mathbf{c}) = \sum_{i=1}^{D} c_i \phi_i(\mathbf{x}) + b \tag{4}$$

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.

$$R(\mathbf{c}) = \frac{1}{N} \sum_{i=1}^{N} \mid y_i - f(\mathbf{x}_i, \mathbf{c}) \mid_\epsilon + \lambda \|\mathbf{c}\|^2 \tag{5}$$

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

$$\mid y_i - f(\mathbf{x}_i, \mathbf{c}) \mid_\epsilon = \max(\mid y_i - f(\mathbf{x}_i, \mathbf{c}) \mid - \epsilon, 0) \tag{6}$$

Vapnik showed in [20] that the function that minimizes the functional in equation (5) depends on a finite number of parameters, and has the following form:

$$f(\mathbf{x}, \alpha, \alpha^*) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) K(\mathbf{x}, \mathbf{x}_i) + b, \tag{7}$$

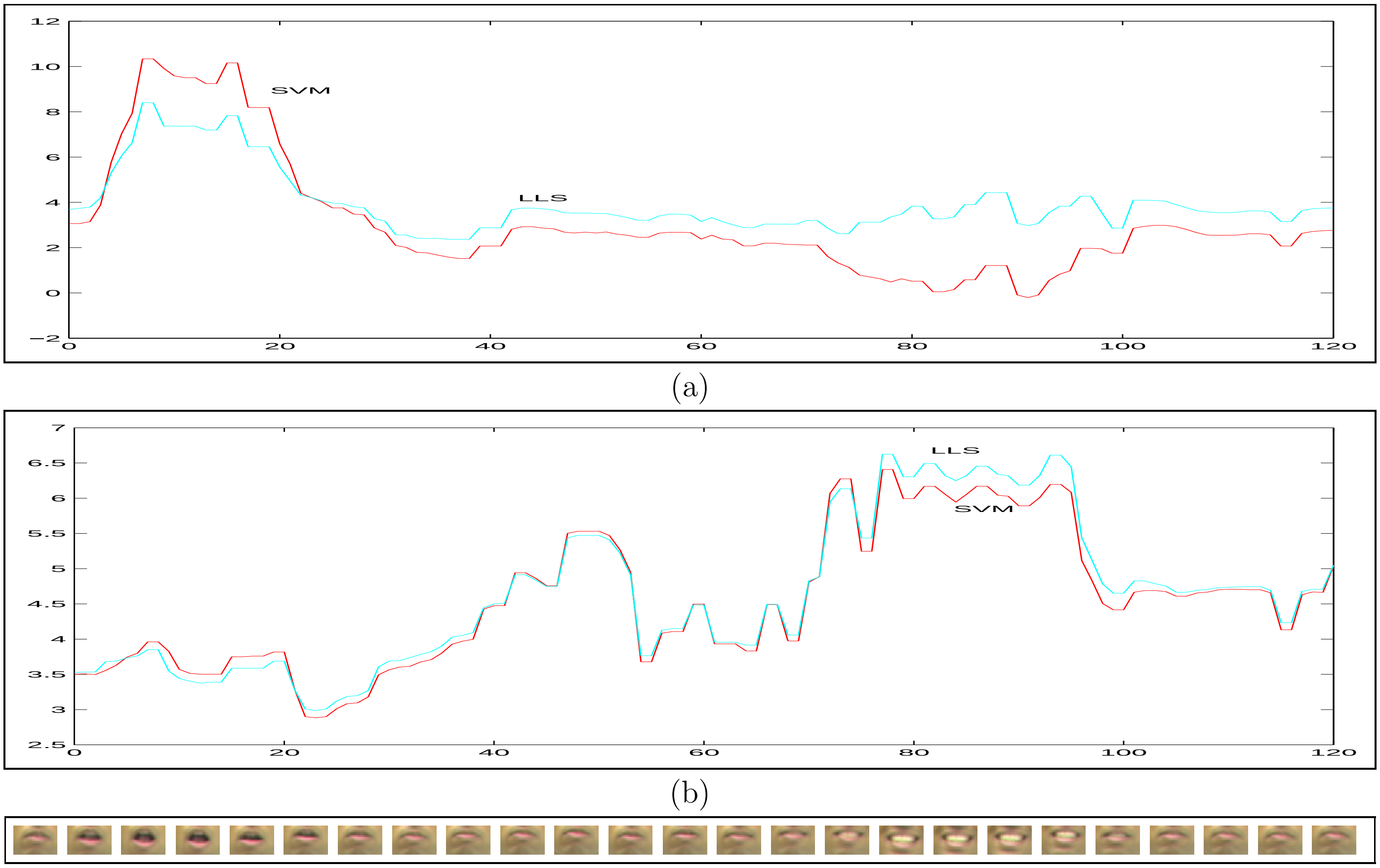
where $\alpha_i^* \alpha_i = 0, \ \alpha_i, \alpha_i^* \geq 0 \quad i = 1, \ldots, N$, and $K(\mathbf{x}, \mathbf{y})$ is the so called *kernel* function, and describes the inner product in the D-dimensional feature space

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

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 (6), we obtain an estimate which is less sensitive to outliers.

4.5 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 4.1. 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 by a median filter of length 3. In figure 5 we present the results of testing this regression technique



(a)

(b)

Figure 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.

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. 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.

5 Conclusions and Future Work

In this paper 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. It also treats expressions as a static phenomenon. Therefore, at this stage the important open questions are: how do we incorporate top-down knowledge that would "specify" what parameters need to be estimated by the bottom-up processes described in this paper? and how do we deal with the dynamic aspects of facial expressions?

6 Acknowledgment

We would like to thank the members of CBCL, Mike Oren, Edgar Osuna, Tony Ezzat and Shayan Mukherjee for discussions and help at various stages of this research. We also request the scientific community to not use this research for military or any other unethical purposes.

References

[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. Image representations for visual learning. *Science*, 272(5270):1905–1909, June 1996.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

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

[12] 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).

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

[14] 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.

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

[16] 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.

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

[18] 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.

[19] 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.

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