Image-Based Stress Recognition Using a Model-Based Dynamic Face Tracking System

Dimitris Metaxas, Sundara Venkataraman, and Christian Vogler

Center for Computational Biomedicine, Imaging and Modeling (CBIM)
Department of Computer Science
Rutgers University
dnm@cs.rutgers.edu
http://www.cs.rutgers.edu/~dnm

Abstract. Stress recognition from facial image sequences is a subject that has not received much attention although it is an important problem for a host of applications such as security and human-computer interaction. This class of problems and the related software are instances of Dynamic Data Driven Application Systems (DDDAS). This paper presents a method to detect stress from dynamic facial image sequences. The image sequences consist of people subjected to various psychological tests that induce high and low stress situations. We use a model-based tracking system to obtain the deformations of different parts of the face (eyebrows, lips, mouth) in a parameterized form. We train a Hidden Markov Model system using these parameters for stressed and unstressed situations and use this trained system to do recognition of high and low stress situations for an unlabelled video sequence. Hidden Markov Models (HMMs) are an effective tool to model the temporal dependence of the facial movements. The main contribution of this paper is a novel method of stress detection from image sequences of a person's face.

1 Introduction

Stress detection of humans has been a well researched topic in the area of speech signal processing [Steeneken, Hansen 99], while very little attention has been paid to recognizing stress from faces. Recognizing stress from faces could complement speech-based techniques and also help in understanding recognition of emotions. The challenges in this domain is that the data from each person are continuous and dynamic and each person expresses stress differently. Therefore the recognition of stress and the associated development of the necessary software is a DDDAS. In the next two sections we illustrate the data collection procedure and the algorithms that will be used for training/recognition.

1.1 Overview of the System

The data used for our experiments were obtained from a psychological study at the University of Pennsylvania. The subjects of the study were put through a battery of

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3038, pp. 813–821, 2004.

tests that induce high and low stress situations. The subjects were videotaped as they took the tests.

A generic model of the face is fitted to the subjects' face and the tracking system is run on the face image sequence with this initial fit of the model in the first frame. The tracking system does a statistical cue integration from computer vision primitives such as edges, point trackers and optical flow. The face model incorporates some parametric deformations that give jaw, eyebrow and basic lip movements. The face tracker gives values for these parametric deformations as a result of the tracking. These are the parameters we will use to learn the movements that correspond to different stress situations. The learning methods will be trained on these parameters and the parameters of a sequence from an unknown stress situation are tested against the learned system to classify a given sequence as a high or low stress condition.

1.2 Learning and Recognition

We will evaluate different learning approaches to train the system for low and high stress conditions. The tracking result will give us parameters which account for the rigid transformations between the face movements and also deformations of the mouth, eyebrow etc. for stress conditions. The recognition will be done using Hidden Markov Models (HMMs). HMMs were chosen since they have been used to model temporal dependence very effectively in American Sign Language (ASL) recognition. Also, we will use the boosting approach to enhance the learning methods and avoid overfitting.

2 Deformable Model Tracking

2.1 Deformable Models

The face tracking system uses a face model that deforms according to movement of a given subject's face. So the shape, position and orientation of the model surface can change. These changes are controlled by a set of n parameters $\mathbf{q}$. For every point i on the surface of the model, there is a function F_i that takes the deformation parameters and finds

$$p_i = F_i(\mathbf{q}) \tag{1}$$

where p_i is the position of the point in the world frame [Metaxas 97].

In addition, computer vision applications, such as deformable model tracking, require the first order derivatives, so we restrict F_i to the class of functions for which the first order derivative exists everywhere with respect to $\mathbf{q}$. This derivative is the Jacobian J_i, where

$$J_i = \begin{bmatrix} | \\ \frac{\partial p_i}{\partial q_1} \ldots\ldots \frac{\partial p_i}{\partial q_n} \\ | \end{bmatrix} \tag{2}$$

Each column of the Jacobian $\mathbf{J}_i$ is the gradient of p_i with respect to the parameter q_i.

2.2 Fitting and Tracking

In principle, there exists a clean and straightforward approach to track deformable model parameters across image sequences. Low-level computer vision algorithms generate desired 2D displacements on selected points on the model, that is differences between where the points are currently according to the deformable model and where they should be according to measurements from the image. These displacements, also called 'image forces', are then converted to *n*-dimensional displacement $\mathbf{f}_g$ in the parameter space, called the *generalized force* and used as a force in the first-order massless Lagrangian system :

$$\dot{\mathbf{q}} = \mathbf{f}_g + F_{\text{internal}}(\mathbf{q}) \tag{3}$$

where $F_{\text{internal}}(\mathbf{q})$ is the result of internal forces of the model (i.e. elasticity of the model, preset). We integrate this system with the classical Euler integration procedure, which eventually yields a fixed point, where $\mathbf{f}_g = \mathbf{0}$. This fixed point corresponds to the desired new position of the model.

In order to use the system of (1), we have to accumulate all the 2D image forces from the computer vision algorithms into $\mathbf{f}_g$. First, we convert each image force $\mathbf{f}_i$ on a point p_i into a generalized force $\mathbf{f}_{gi}$ in parameter space, which describes the effect that single displacement at point p_i has on all the parameters. Obtaining a generalized force $\mathbf{f}$g then simply consists of summing up all $\mathbf{f}_{gi.:}$

$$\mathbf{f}_g = \sum_i \mathbf{f}_{gi} \quad \text{where} \quad \mathbf{f}_{gi} = \sum_i \mathbf{B}_i^T \mathbf{f}_i \tag{4}$$

and

$$\mathbf{B}i = \frac{\partial \mathbf{Proj}}{\partial p}\Big| \; pi \, \mathbf{J}i \tag{5}$$

$\mathbf{B}_i$ is the projection of the Jacobian $\mathbf{J}_i$ from world coordinates to image coordinates via the projection matrix **Proj** at point p_i.

Generating the generalized force this way works fine as long as all the image forces come from the same cue (diff. algorithms on the same image). When there are multiple cues from multiple vision algorithms, combining the cues becomes a hard problem. In order to effectively combine them statistically we will need to know the

distributions of the individual cues ahead of time, but it is hard to estimate these distributions beforehand.

We choose the framework of affine regions [Goldenstein et al. 2001, 2003] to estimate the distributions of the cues within small regions and apply the equivalent of the central limit theorem to these affine regions to make a Gaussian approximation. Then we use maximum likelihood estimation to get the final generalized force.

The face model itself was made from a publicly available geometric model of the head, available from the University of Washington as part of [Pighin et al.99]. A face mask was cut out of this original model and we obtained a static model with 1,100 nodes and 2000 faces. Then, parameters and associated regions are defined for the raising and lowering of eyebrows, for the smiling and stretching of the mouth, for the opening of the jaw as well as the rigid transformation parameters for the model frame.

2.3 Improvements of the Face Model

In this version of the system we have added asymmetric deformations for the eyebrows and the mouth region i.e. the left and right eyebrows, the left and right ends of the lips of the mouth are no longer tied together. This is essential since one of the major indicators of stress is asymmetric lip and eyebrow movements and the original framework did not support that.

The deformation parameters all put together form about 14 parameters. The tracking results from a particular video sequence will give us these 14 parameters for the model for each time instance. We perform these tracking experiments on various subjects and use the parameters we obtain to train the HMMs.

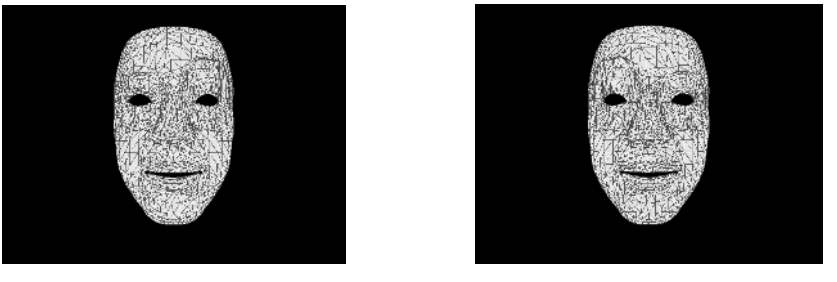

Fig. 1. Left eyebrow movement in the model and right eyebrow movement in the model

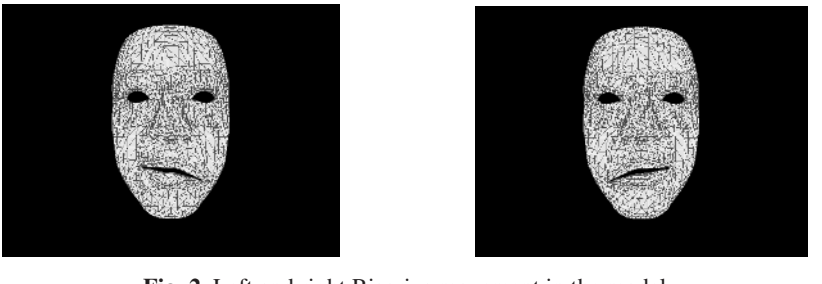

Fig. 2. Left and right Risorius movement in the model

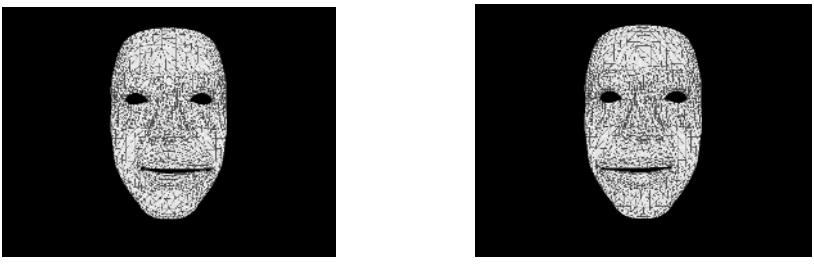

Fig. 3. Left and right lip stretching in the model

3 Stress Recognition

The computational detection of a human undergoing a stress reaction can be broken up into two stages. The first stage consists of recognizing the possible individual displays of stress response in the human face such as eye movements and blinking, various negative facial expressions. The second stage consists of accumulating the information collected in the first stage and deciding whether these displays occur frequently enough to classify as a stress response.

From an abstract point of view, the first stage corresponds to the task of detecting specific patterns in a time-varying data signal. Depending on the specific task and the pattern that we are looking for, the signal often simply consists of the deformable model parameter vectors that we estimate during the face tracking process. For instance, in order to detect rapid head movements, we are interested in the rigid body component of the parameter vector. The orientation, position, and derivatives thereof contain all the necessary information for this particular pattern. Likewise, in order to detect specific negative facial expressions, we are interested in the nonrigid deformation component of the parameter vector, which controls the eyebrow and mouth movements, and nearby regions.

Eye blinking is slightly more complicated to handle, because the deformable model does not contain any parameters that control it directly. However, the output of the tracking algorithm does contain information on the location of the eyes in the human face at any given point in time. The eyes themselves are represented as holes in the deformable model, delimited by the region that is formed by a set of nodes. Because from the deformable model parameters the position of these nodes can be deduced, it is possible to project their positions into image space, and thus find out which region in the video frame corresponds to the eyes. We then use a grayscale level averaging method on the region in the video frame to determine the degree, to which the eyes are opened or closed - uniform grayscale levels indicate that the eyelids are covering the irises, whereas more diverse grayscale levels indicate that the irises are visible. Just like the model parameters, the degree of openness of the eyes can thus be quantified in a few numbers, which is important for the subsequent training of the recognition algorithm.

Detecting stress patterns in a time-varying signal is very similar to other well-known activity recognition tasks, such as gesture recognition, and sign language recognition. For these tasks, hidden Markov models (HMMs) have been shown to be highly suitable, for several reasons: First, the HMM recognition algorithm (Viterbi

decoding) is able to segment the data signal into its constituent components implicitly, so it is not necessary to concern ourselves with the often extremely difficult problem of segmenting a data signal explicitly. Second, the state-based nature of HMMs is a natural match for the task of recognizing signals over a period of time. Third, the statistical nature of HMMs makes them ideal for recognizing tasks that exhibit an inherent degree of variation; for example, due to motor limitations, humans generally do not perform the same movement twice in exactly the same way, even if they intend to. To make HMM-based recognition of potentially stress-related displays work, we first train the HMMs on hand-labeled examples of such displays. The labeled examples include information on the starting and ending frame of the display, as well as the class into which it belongs: a specific type of negative facial expression, rapid head movement, eye blinking, and so on. Then, during the recognition phase, the HMM algorithm detects from the tracking data which ones of these types of displays occur in the video, and when; that is, at which frames.

4 Hidden Markov Models

4.1 Background

Hidden Markov Models (HMMs) have been a very effective tool in capturing temporal dependencies in data and fitting them to models. They have been applied very effectively to the problem of American Sign Language (ASL) recognition and to speech recognition in particular with reasonable commercial success.

Hidden Markov models are a type of statistical model embedded in a Bayesian framework. In their simplest form, an HMM λ consists of a set of N states S_1, S_2, $\ldots S_N$. At regularly spaced discrete time intervals, the system transitions from state S_i to S_j with probability a_{ij}. The probability of the system initially starting in state S_i is $\prod_i$.

Each state S_i generates output $O \in \Omega$, which is distributed according to a probability distribution function $b_i(O) = P\{\text{Output is } O \mid \text{System is in } S_i\}$. In most recognition applications $b_i(O)$ is actually a mixture of Gaussian densities.

In most applications, the HMMs correspond to specific instances of a situation (in our case, different high stress situations). Then the recognition problem is reduced to find the most likely state sequence through the network. So we would like to find a state sequence $Q = Q_1,\ldots,Q_T$ over an output sequence $O = O_1,\ldots,O_T$ of T frames, such that $P(Q,O|\lambda)$ is maximized. So we have

$$\delta_t(i) = \max_{Q1,\ldots,Qt-1} P(Q_1 Q_2 \ldots Q_t = S_i, O \mid \lambda) \tag{6}$$

and by induction

$$\delta_{t+1}(i) = b_i(O_{t+1}) \cdot \max_{1 \le j \le N} \{\delta_t(j) a_{ji}\} \tag{7}$$

$$P(Q, O \mid \lambda) = \max_{1 \le i \le N} \{\delta_T(i)\}$$

The Viterbi algorithm computes this state sequence in $O(N^2T)$ time, where N is the number of states in the HMM network. The Viterbi algorithm implicitly segments the observation into parts as it computes the path through the network of HMMs.

A more detailed introduction and description of algorithms and inference in HMMs is described in [Rabiner 89].

5 Experiments and Results

The data for the experiments were collected at the University of Pennsylvania NSBRI center. The subjects of the experiment were put through the neurobehavioral test battery (NTB), while being videotaped. The NTB tests consist of two sessions. Session I : The subjects perform the 'Stroop word-color inference task (Stroop). This requires the subject to filter out meaningful linguistic information whereby subjects must respond with the printed color name rather than the ink color name. During the task it is difficult to ignore these conflicting cues and automatic expectations that are associated with impulse control. The Psychomotor Vigilance Task (PVT) is a simple, high-signal-load reaction time test designed to evaluate the ability to sustain attention and respond in a timely manner to salient signals [Dinges 85]. The probed recall memory (PRM) test controls report bias and evaluates free working memory [Dinges 93]. Descending subtraction task (DST) requires the subject to perform serial subtractions of varying degrees of difficulty. Visual motor task (VMT) requires subjects to remember and replicate positions of a continually increasing sequence of flashing blocks and Digit symbol substitution task (DSST) assesses cognitive speed and accuracy trade-offs.

Session II : Serial addition subtraction task (SAST) assesses cognitive throughput (speed and accuracy trade-offs),. Synthetic workload (SYNW) task is a multi-cognitive task comprising four tasks completed simultaneously on a split screen, including probed memory, visual monitoring and simple auditory reaction time. Meter reading task (MRT) is a numerical memory and recall task; Working memory task (WMT) requires the subjects to determine whether the target stimulus is the same or different from a previously displayed cue stimulus; Logical reasoning task (LRT) involves attention resources, decision-making and response selection; Haylings sentence completion task (HSC) involves subjects completing a sentence with a single word that is either congruous or incongruous with the overall meaning of the sentence.

The Stroop, HSC and DST are verbal tasks, while PVT, PRM, VMT, SAST, SYNW, MRT, WMT, LRT and DSST are non-verbal tasks, requiring computer inputs as responses.

Workload demands are initially low, and increase across the performance bout, so that in the second half subjects are performing under high workload demand, designed to induce high behavioral distress. Periodically throughout the test bout, onscreen feedback is provided. During the low workload periods, this feedback is positive. As the workload demands increase, the feedback is negative. The feedback is identical for all subjects, and is independent of performance levels. Additionally, during the high workload portions of the test bout, uncontrolled and unexpected "computer failures" occur that are experimenter generated. The obtained video sequences were

classified into high and low stress conditions and blind video sequences on which testing needs to be done.

The following figures show examples of high and low stress situations from the tests. The images are normalized i.e. the inverse of the rigid transformations inferred from the deformable model tracking are applied to the face model to produce a frontal face with the image from the video sequence texture mapped onto this model.

From the sequences that were analyzed, we found that some of the major indicators of high stress were eyebrow movements, asymmetric lip deformations and baring of teeth. The tracking results from such sequences were used as input to the HMM learner.

The experiments were conducted on 25 datasets in all with approximately one half of the data being high stress sequences and the other half low stress sequences.

We used two ergodic HMMs one for low and one for high stress conditions. The feature vector we used for training the HMMs were the face asymmetries (difference between the left and right eyebrows, Risorius and lip stretching deformation parameters. The training and testing was independent of the subjects who were part of the experiment.

Recognition was performed with different amounts of data splits. A 75% - 25% data split between training and test data respectively gave us results where all the 6 datasets in the test sets were correctly identified as low/high stress conditions. A 50% - 50% split between the training and test data sets gave us results where 12 out of the 13 samples were correctly identified as low/high stress conditions.

Fig. 4. (a) Low stress condition, (b) high stress condition (asymmetric lip deformation), and (c) high stress condition (baring of teeth)

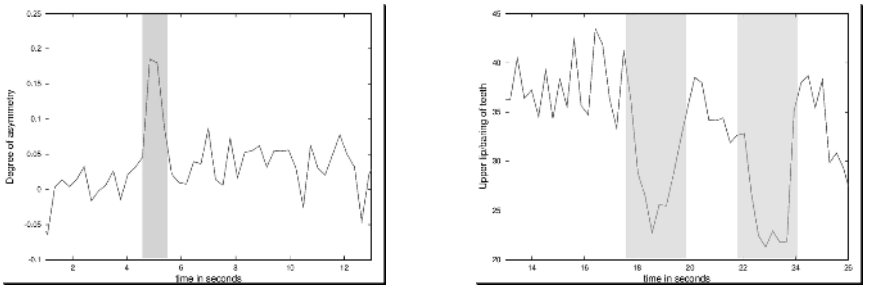

Fig. 5. Asymmetry in lips (indicator of high stress) and baring of teeth (indicator of high stress)

6 Conclusions

In this paper we have presented a DDDAS for the recognition of stress from facial expressions. Our method is based on the use of deformable models and HMMs that can deal with the dynamically varying data and variances in the expression of stress among people.

Acknowldedgements. This work has been funded by an NSF-ITR 0313184 to the first author.

References

[1] Steeneken, H. J. M., Hansen, J. H. L. (1999). *Speech under stress conditions: Overview of the effect on speech production and on system performance.* IEEE International Conference on Acoustics, Speech and Signal Processing.

[2] Goldenstein, S., Vogler C., Metaxas D.N., (2001). *Affine arithmetic based estimation of cue distributions in deformable model.*

[3] Goldenstein, S., Vogler C., Metaxas D.N., (2003). *Statistical Cue Integration in DAG Deformable Models.* IEEE Transactions on Pattern Analysis and Machine Intelligence, July 2003.

[4] Vogler, C., Metaxas, D. N. (1999). *Parallel Hidden Markov Models for American Sign Language recognition.* International Conference on Computer vision, Kerkyra, Greece.

[5] Pighin, F., Szeliski, R., Salesin, D., (1999). *Resynthesizing Facial animation through 3D Model-based tracking*. International Conference on computer vision.

[6] Rabiner, L. R., (1989). *A tutorial on hidden Markov models and selected applications in speech recognition.* Proceedings of the IEEE, Vol. 77.

[7] Metaxas, D. N., (1997). *Physics-based deformable models: Applications to computer vision, graphics and medical imaging.* Kluwer Academic Press.