Face Model Adaptation for Tracking and Active Appearance Model Training

F. Dornaika
Linköping University
SE-581 83 Linköping, Sweden
dornaika@isy.liu.se

J. Ahlberg
Swedish Defence Research Agency (FOI)
SE-581 11 Linköping, Sweden
jorahl@foi.se

Abstract

In this paper, we consider the potentialities of adapting a 3D deformable face model to video sequences. Two adaptation methods are proposed. The first method computes the adaptation using a locally exhaustive and directed search in the parameter space. The second method decouples the estimation of head and facial feature motion. It computes the 3D head pose by combining: (i) a robust feature-based pose estimator, and (ii) a global featureless criterion. The facial animation parameters are then estimated with a combined exhaustive and directed search. Tracking experiments and performance evaluation demonstrate the feasibility and usefulness of the developed methods. These experiments also show that the proposed methods can outperform the adaptation based on a directed continuous search.

Keywords: *3D deformable models, face model adaptation, face tracking, active appearance models, analysis-by-synthesis approaches, human-computer interaction, learning*

1 Introduction

Face and facial feature tracking in a video sequence has been recognized as an essential prerequisite for robust facial expression/emotion analysis, face recognition, and model-based image coding. This information is also a very important primitive for smart environments and perceptual user interfaces where the poses and movements of body parts need to be determined. With the wide availability of inexpensive cameras and increasingly better support of streaming video by computers, vision-based head and facial motion tracking techniques are well justified. The issue of face recognition and facial analysis has been extensively addressed in recent years [11]. Different approaches including eigenfaces [10], elastic graph models [7], deformable templates [12], Active Shape Models (ASMs) [4] and Active Appearance Models (AAMs) [3] have shown to be promising under different assumptions.

A huge research effort has been devoted to detecting and tracking of head and facial features in 2D and 3D (e.g., [9, 12]). Most tracking approaches take advantage of the

BMVC 2003 doi:10.5244/C.17.57

constrained scenario: the face and/or the head are not viewed as arbitrary tracked objects. A model-based approach is favored.

In [2] , we have formulated the problem of real-time face and facial feature tracking/adaptation using an Active Appearance Model search. The search algorithm employs a directed continuous optimization technique, and it belongs to the class of analysis-by-synthesis approaches. Applying an AAM search requires the prior knowledge of two key elements: (i) a statistical facial texture model (e.g., using a Principal Component Analysis method), and (ii) a gradient matrix which encodes the differential relationship between the parameter space and the texture differences. To recover these elements, the 3D model should be adapted to a set of facial images (training set) either manually or semi-automatically. However, the sheer number of training images provided by video sequences makes the manual and semi-automatic adaptations very tedious. Moreover, the automatic adaptation based on a directed continuous optimization (AAM search) may suffer from some inaccuracies possibly caused by bad convergence behaviours.

In this paper, we propose two novel adaptation methods that can be used for face tracking and AAM training. These methods do not employ any directed continuous search (only a rough facial texture model is required), and are able to tackle the above disadvantages. The first method computes the adaptation by adopting a locally exhaustive and directed search in the parameter space. The second method decouples the estimation of head and facial feature motion. It computes the 3D head pose and facial motion in sequence. The 3D head pose is estimated by combining: (i) a robust feature-based pose estimator, (ii) a global featureless criterion. The facial animation parameters are then estimated with a combined exhaustive and directed search. The remaining of the paper is organized as follows. Section 2 introduces the deformable face model. Section 3 describes the real-time directed continuous search. Section 4 describes the two proposed adaptation methods. Section 5 presents some experimental results as well as a performance study.

2 A deformable face model

2.1 A parameterized 3D face model

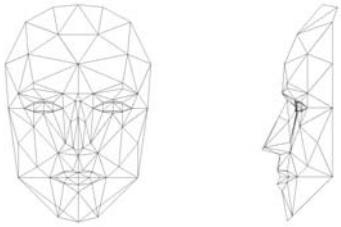


Figure 1: Candide-3 with 113 vertices and 183 triangles.

Building a generic 3D face model is a challenging task. Indeed, such a model should account for the differences between different specific human faces as well as between different facial expressions. This modelling was explored in the computer graphics, computer vision, and model-based image coding communities. Many different face models

exist, varying significantly in complexity. In our study, we use the 3D face model *Candide*, developed at Linköping University in the eighties [1]. This 3D wireframe model is a relatively simple model, see Figure 1, consisting of approximately 113 vertices and 183 triangles, and even though more complex models are preferred by face animators and computer graphics people, *Candide* will suit our needs here.

Thus, the shape up to a global scale can be fully described by the $3n$-vector $\mathbf{g}$–the concatenation of the 3D coordinates of all vertices $\mathbf{P}_i$. The vector $\mathbf{g}$ can be written as:

$$\mathbf{g} = \overline{\mathbf{g}} + \mathbf{S}\,\sigma + \mathbf{A}\,\alpha \tag{1}$$

where $\overline{\mathbf{g}}$ is the standard shape of the model, and the columns of $\mathbf{S}$ and $\mathbf{A}$ are the Shape and Animation Units, respectively. The Shape Units provide a way to deform the 3D mesh such as to make the head wider, the eyes wider, etc. The Action Units provide a way to deform the 3D mesh according to some predefined facial animations [5]. Thus, the term $\mathbf{S}\,\sigma$ accounts for shape variability (inter-person variability) while the term $\mathbf{A}\,\alpha$ accounts for the facial animation (intra-person variability).

The adopted projection model is the weak perspective projection model. Therefore, the mapping between the 3D face model and the image is given by a 2×4 matrix $\mathbf{M}$. Thus a 3D vertex $\mathbf{P}_i = (X_i, Y_i, Z_i)^T \subset \mathbf{g}$ will be projected onto the image point $\mathbf{p}_i = (u_i, v_i)^T$ given by:

$$(u_i, v_i)^T = \mathbf{M}\,(X_i, Y_i, Z_i, 1)^T \tag{2}$$

The matrix $\mathbf{M}$ encapsulates the camera intrinsic parameters and the 3D head pose represented by a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t} = (t_x, t_y, t_z)^T$. Retrieving the 3D pose parameters ($\mathbf{R}$ and $\mathbf{t}$) from $\mathbf{M}$ is straightforward. We represent the rotation matrix $\mathbf{R}$ by the three Euler angles. Therefore, the geometry of the model is parameterized by the parameter vector $\mathbf{b}$:

$$\mathbf{b} \quad = \quad [\theta_x, \quad \theta_y, \quad \theta_z, \quad \lambda\, t_x, \quad \lambda\, t_y, \quad \lambda\, t_z, \quad \sigma^T \quad \alpha^T \;]^T$$

For a given person, only the 3D head pose and the animation parameters, α, are time dependent.

2.2 Geometrically normalized facial images

A face texture is represented as a geometrically normalized image. The geometry of this image is obtained by projecting the standard shape $\overline{\mathbf{g}}$ (wireframe) using a standard 3D pose (frontal view) onto an image with a given resolution (intrinsic parameters). This geometry is represented by a triangular 2D mesh. The texture of this geometrically normalized image is obtained by texture mapping from the triangular 2D mesh in the input image using piece-wise affine transform. For a very fast texture mapping (image warping), we have exploited the fact that the 2D geometry of the destination mesh can be known in advance. In fact, the geometrical normalization normalizes three different things: the global motion, the local motion, and the geometrical differences between individuals. Figure 2 displays the geometrical normalization results associated with two input images (256×256) having a correct adaptation. The geometrically normalized images are of resolution 40×42. Computing the geometrically normalized image of resolution 40×42 takes about 0.3 ms on a 1500 MHz PC.

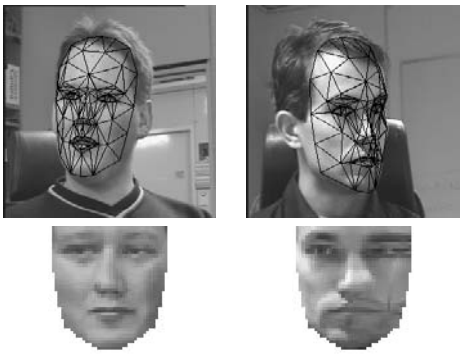


Figure 2: Two input images with correct adaptation (top). The corresponding geometrically normalized image (bottom).

2.3 Texture consistency

The texture $\mathbf{x}$ of any geometrically normalized image is given by (using Principal Component Analysis method):

$$\mathbf{x} \approx \overline{\mathbf{x}} + \mathbf{X}\,\xi \tag{3}$$

where $\overline{\mathbf{x}}$ is the mean texture, the orthogonal columns of $\mathbf{X}$ are the texture modes (eigenfaces) and ξ is the vector of texture parameters.

Assume that we have a face image I whose model parameters are given by the vector $\mathbf{b}$. Using these parameters, the geometrically normalized image can be easily reconstructed. Let $\mathbf{x}$ be the resulting texture. This texture can be approximated by the texture modes according to:

$$\begin{aligned} \xi &= \mathbf{X}^T(\mathbf{x}-\overline{\mathbf{x}}) \\ \hat{\mathbf{x}} &= \overline{\mathbf{x}} + \mathbf{X}\mathbf{X}^T(\mathbf{x}-\overline{\mathbf{x}}) \end{aligned}$$

where $\hat{\mathbf{x}}$ is the best approximate of $\mathbf{x}$ in the space represented by the texture modes.

If the model geometry, $\mathbf{b}$, is a good fit to the image I, then the residual error between the texture $\mathbf{x}$ and its approximated texture $\hat{\mathbf{x}}$ is small since the remapped texture will be consistent with the statistical model of a face texture.

Thus, a reliable measure of the goodness of any fit, $\mathbf{b}$, can be given by the norm of the associated residual image between the remapped texture and its PCA approximation:

$$e(\mathbf{b}) = \|\mathbf{r}\|^2 = \|\mathbf{x}(\mathbf{b}) - \hat{\mathbf{x}}(\mathbf{b})\|^2 \tag{4}$$

3 The adaptation problem

Given an image of a face (or a video sequence), the adaptation consists of estimating the model parameters $\mathbf{b}$ for each image. For a given person, σ is constant and can be estimated using the method described in [8]. In this case, $\mathbf{b}$ can be simplified to:

$$\mathbf{b} = [\,\theta_x,\ \theta_y,\ \theta_z,\ \lambda t_x,\ \lambda t_y,\ \lambda t_z,\ \alpha^T\,]^T$$

Thus for a given sequence, the vector $\mathbf{b}$ describes the time dependent geometry of the 3D wireframe model (3D head pose and facial animation). In a tracking context, the adaptation results associated with the current frame will be handed over to the next frame.

Without loss of generality, we have chosen the following Action Units [5]: 1) Jaw drop, 2) Lip stretcher, 3) Lip corner depressor, 4) Upper lip raiser, 5) Eyebrow lowerer, 6) Outer eyebrow raiser (i.e. the dimension of the vector α is 6). These Action Units are enough to cover most common facial expressions (mouth and eyebrow movements).

4 Adaptation with a directed continuous search

According to the Active Appearance Model search paradigm, the model parameters are estimated such that the texture of the corresponding geometrically normalized image is as close as possible to the face space [2]. In other words, the model geometry, $\mathbf{b}$, is estimated by locally minimizing the residual error (Eq. (4)). The active appearance model search employs a simple gradient descent method in the following manner. For a starting value of $\mathbf{b}$, supposed to be close to optimum, we compute $\mathbf{r}(\mathbf{b})$ and $e(\mathbf{b})$, and find an update vector $\Delta\mathbf{b}$ by multiplying the residual image with an update matrix:

$$\Delta\mathbf{b} = -\mathbf{G}^{\dagger}\,\mathbf{r} = -(\mathbf{G}^{T}\,\mathbf{G})^{-1}\,\mathbf{G}^{T}\,\mathbf{r} \tag{5}$$

where $\mathbf{G} = \frac{\partial \mathbf{r}}{\partial \mathbf{b}}$ is the gradient matrix associated with the residual error $\mathbf{r}$, and $\mathbf{G}^{\dagger}$ is the pseudo-inverse of the gradient matrix. The vector $\Delta\mathbf{b}$ gives a displacement in the parameter space allowing to decrease the error. We compute a new parameter vector and a new error:

$$\begin{aligned} \mathbf{b}' &= \mathbf{b} + \mu\,\Delta\mathbf{b} \\ e' &= e(\mathbf{b}') \end{aligned} \tag{6}$$

If $e' < e$, we update $\mathbf{b}$ according to Eq. (6) and the process is iterated until convergence. If $e' \geq e$, we try small update steps having the same direction. Convergence is declared when the error cannot be improved anymore. The gradient matrix $\mathbf{G} = \frac{\partial \mathbf{r}}{\partial \mathbf{b}}$ is considered approximately fixed, i.e. $\mathbf{r}(\mathbf{b})$ is supposed to be a linear function of the parameter vector $\mathbf{b}$. It is estimated once from the training set where the model was correctly adapted to each example. It is created by numeric differentiation, systematically displacing each parameter and computing an average over the training set.

5 Combined locally exhaustive and directed search

It is clear that the directed continuous search has the advantage that the adaptation can run in real-time. On the other hand, it has several disadvantages. First, the gradient matrix should be computed/approximated in advance. Second, the head motion between two successive frames should be relatively very small. Third, there is a risk that the directed search may end at non-desired local minima, which may lead to non-accurate adaptation especially for the out-of-plane motions.

To overcome the disadvantages of the directed continuous search, we propose two different methods that only require a rough facial texture model. The first method can be utilized offline for the purpose of motion analysis and Active Appearance Model training. The second one can be used for on-line tracking as well.

5.1 First method

We approach the minimization problem differently. We use the following combined exhaustive and directed search in the parameter space. Without loss of generality, we assume that the dimension of the vector $\mathbf{b}$ is 12. The minimization consists of two successive stages which are utilized by each frame in the video sequence:

1. *Exploration stage.* At each step $t = 1, 2, \cdots, T$, the locally best parameter, $\mathbf{b}_j^{[t]}$, is chosen by changing each parameter $i \in \{1, \cdots, 12\}$ under fixed values, $[\mathbf{b}_k^{[t-1]} : k \neq i, ; k \in \{1, \cdots, 12\}]$, of other parameters. The choice yields the largest decrease of the error $e(\mathbf{b}^{[t]})$ with respect to $e(\mathbf{b}^{[t-1]})$ providing that the parameters differ by only the value of the locally best parameters $\mathbf{b}_j^{[t]}$. The exploration steps are repeated while the error decreases further. For each step and for each parameter, the exploration locally exhausts a given number of the equispaced parameter values within the range $[\mathbf{b}_i^{[t-1]} - \Delta_i/2, \mathbf{b}_i^{[t-1]} + \Delta_i/2]$. The parameters $\mathbf{b}^{[0]}$ are set to the values computed at the previous frame. In our implementation, Δ_i is set to 12 degrees for the rotation angles and to 13 pixels for the 2D translation. The number of equispaced values is set to 10 or 20.

2. *Search stage.* The exploration steps converge to a final local minimum value $e(\mathbf{b}^{[T]})$, and the vector $\mathbf{b}^{[T]}$ allows for inferring the possible steepest descent direction in the parameter space. The search along this direction refines further the obtained parameters $\mathbf{b}_\mu = \mathbf{b}^{[T]} + \mu\,(\mathbf{b}^{[T]} - \mathbf{b}^{[0]})$. Once the model geometry is found, it is handed over to the next frame.

The CPU-time of the adaptation based on the above scheme is proportional to the number of times the error function is evaluated. This number is given by $n = \dim(\mathbf{b}) \times N \times T + n_{refi}$ where N is the number of the utilized equispaced values for each parameter, T is the number of steps needed for convergence, and n_{refi} is the number of sampled values used by the refinement stage. For example, when $\dim(\mathbf{b}) = 12$, $N = 11$, $T = 4$ (on average 4 steps are needed for convergence), and $n_{refi} = 10$ then a typical n should be 538 evaluations per frame. Each evaluation takes about 0.6 ms on a 1500 MHz PC (0.3 ms for image warping, and 0.3 ms for PCA approximation). Thus, adapting one frame may take 322 ms which cannot allow for a real-time tracking.

To reduce the adaptation CPU-time, one should reduce the CPU-time associated with the evaluation itself and/or the number of the evaluations, n.

Reducing the evaluation CPU-time by modifying the error function For a given model geometry (represented by the vector $\mathbf{b}$), the error is given by:

$$e(\mathbf{b}) = \|\mathbf{x}(\mathbf{b}) - \hat{\mathbf{x}}(\mathbf{b})\|^2$$

There are two main tasks involved in this evaluation: (i) image warping, and (ii) PCA approximation. Our idea is to replace the synthesized image, $\hat{\mathbf{x}}(\mathbf{b})$, with the one computed at the previous frame. Therefore, the error becomes :

$$e(\mathbf{b}) = \|\mathbf{x}(\mathbf{b}) - \hat{\mathbf{x}}_{f-1}\|^2 \tag{7}$$

Using this error function, the task of PCA approximation is skipped, and only the task of image warping remains leading to a CPU-time earning that is proportional to the number

of texture modes (eigenfaces). Thus, the CPU-time associated with the evaluation is now reduced to 0.3 ms. Note that the PCA approximation is carried out once per frame once the current geometry is estimated. Therefore, by changing the error function, the CPU-time of the above scheme can be reduced to 161 ms.

5.2 Second method (decoupled adaptation)

In addition to the use of the modified error function (Eq. (7)), the second method utilizes a reduced number of geometrical parameters. For this purpose, we only estimate the animation parameters by a combined exhaustive and directed search. In other words, we decouple the head motion (global motion) from the facial motion (local motion). The head pose estimation exploits the previous adaptation together with previous frame. The head pose is estimated using a RANdom SAmpling Consensus (RANSAC) technique [6] combined with the texture consistency measure (Eq. 4) to avoid drifting. Once the 3D head pose is estimated, the facial animation, α, is estimated using the scheme described in Section 5.1.

Therefore, using the second method the adaptation of one frame can be achieved in 100 ms, which can be considered as suitable for a real-time or near real-time tracking. The whole adaptation process is summarized in Figure 3.

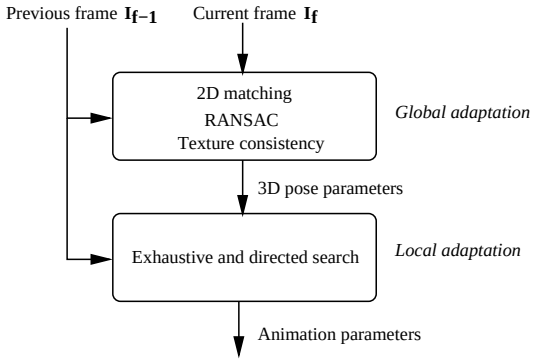


Figure 3: Different steps of the second method (decoupled adaptation).

6 Experiments and accuracy evaluation

To test the proposed methods, we use a training set consisting of 330 images of six different persons from different angles and with different facial expressions. Using this set a texture model has been built using the PCA method. Figure 4 shows the adaptation of a test sequence of a previously unseen person (140 frames) using the proposed locally exhaustive and directed search (Section 5.1). Figure 5 shows the adaptation of a test sequence of 340 frames using the decoupled adaptation (Section 5.2).

Figure 6.**(a)** displays the adaptation errors associated with a synthesized sequence of 250 frames (known ground truth). The solid curves depict the adaptation errors associated with the directed continuous method (AAM search described in Section 4). The dashed curves depict the adaptation errors associated with the combined exhaustive and directed

search (Section 5.1). In this figure, only the three rotations and the eyebrow raiser parameter are depicted since a similar behaviour has been obtained with the other geometrical parameters. For each frame in the synthetic sequence and for each geometrical parameter, the error is the absolute value of the difference between the estimated value and its ground truth value used in animating the synthetic images.

Figure 6.**(b)** displays the same adaptation errors as Figure 6.**(a)** except that the first method has been replaced by the second method (Section 5.2) , i.e., the decoupled adaptation method (dash-dotted curves). As can be seen, the two proposed adaptation methods are more accurate than the directed continuous search. Also, the first method (the slowest one) seems to be slightly more accurate than the second method.

7 Conclusion

We have proposed two methods for model-based face adaptation. The first method can be used for building accurate Active Appearance Models. The second one can be used for the same task as well as for real-time tracking. Both methods only require a simple facial texture model and exploit no prior information about a gradient matrix. The proposed schemes allow one to take into account the multi-modal character of the error function. Moreover, the proposed schemes offer a lot of flexibility as to how many facial animation parameters can be used at running time, which cannot be done with a directed continuous technique where the geometrical parameters are fixed by the training stage.

Figure 4: Adaptation based on the exhaustive and directed search using the first test sequence (140 frames).

Figure 5: Decoupled adaptation using the second test sequence (340 frames).

References

[1] J. Ahlberg. CANDIDE-3 - an updated parametrized face. Technical Report LiTH-ISY-R-2326, Department of Electrical Engineering, Linköping University, Sweden, 2001.

[2] J. Ahlberg. An active model for facial feature tracking. *EURASIP Journal on Applied Signal Processing*, 2002(6):566–571, June 2002.

[3] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *Proc. European Conference on Computer Vision*, pages 484–498, 1998.

[4] T.F. Cootes, C.J. Taylor, D. Cooper, and J. Graham. Active shape models–their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[5] P. Ekman and W.V. Friesen. *Facial Action Coding System*. Consulting Psychology Press, Palo Alto, CA, USA, 1977.

[6] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communication ACM*, 24(6):381–395, 1981.

[7] M. Lades, J.C. Vorbrüggen, J. Buhmann, J. Lange, C.V.D. Malsburg, R. Würtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. on Computers*, 42(3):300–310, March 1993.

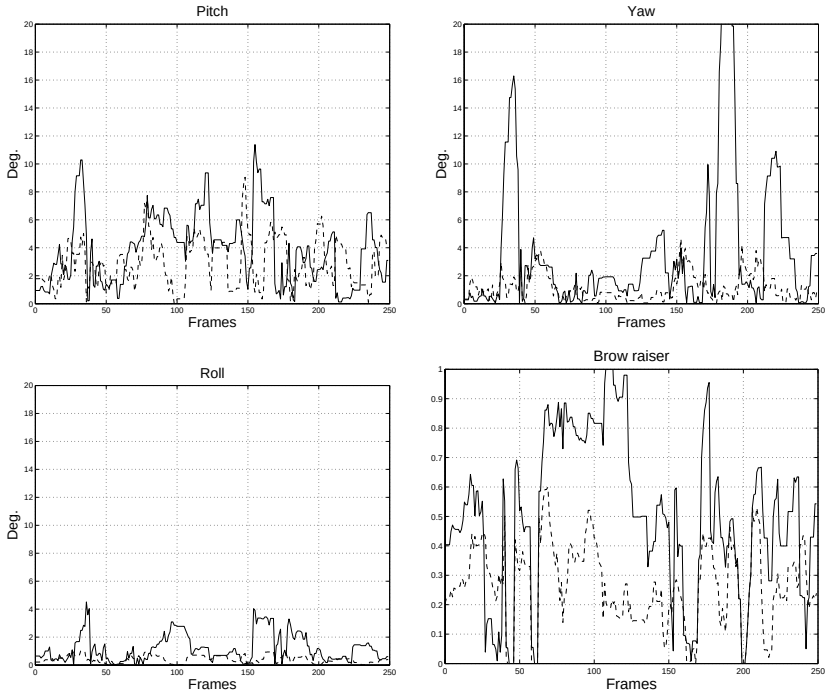
[8] L. Lu, Z. Zhang, H.Y. Shum, Z. Liu, and H. Chen. Model- and exemplar-based robust head pose tracking under occlusion and varying expression. In *Proc. IEEE Workshop on Models versus Exemplars in Computer Vision, (CVPR'01)*, 2001.

[9] J. Ström. Model-based real-time head tracking. *EURASIP Journal on Applied Signal Processing*, 2002(10):1039–1052, 2002.

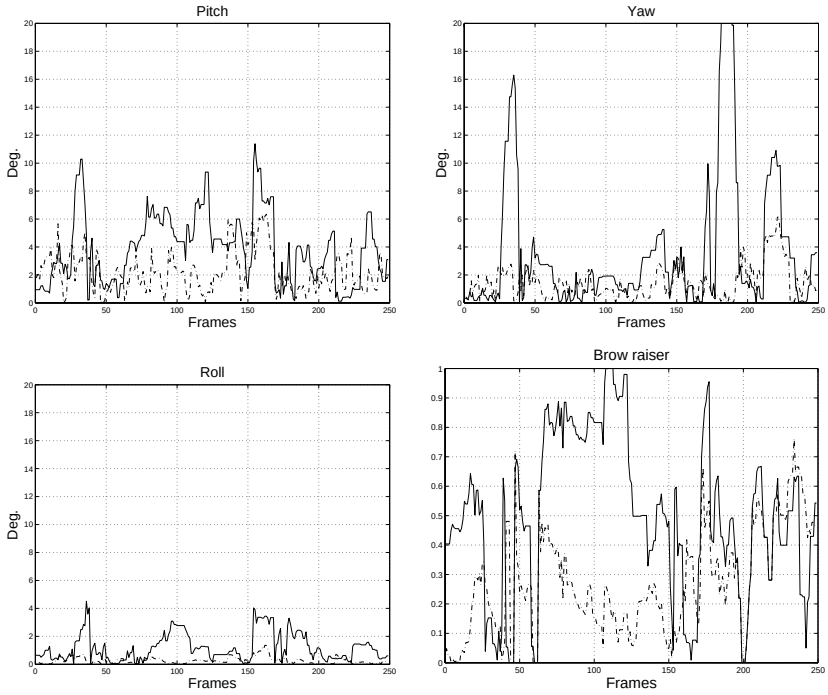
[10] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991.

[11] M.H. Yang, D.J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

[12] A.L. Yuille, D.S. Cohen, and P. Hallinan. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–112, 1992.



(a) Directed continuous search vs. our first method (see text).



(b) Directed continuous search vs. our second method (see text).

Figure 6: Adaptation errors using a synthetic sequence of 250 frames.