A Probabilistic Hierarchical Framework for Expression Classification

Lukasz Zalewski and Shaogang Gong*
*Department of Computer Science, Queen Mary, University of London
London, E1 4NS, UK
[lukas|sgg]@dcs.qmul.ac.uk

Abstract

We address the issue of understanding facial expressions through statistical modelling and analysis without the need for any temporal information whatsoever. We introduce a hierarchical decomposition of a human face into different subcomponents where each of them is modelled using Probabilistic PCA (PPCA). Classification is performed by fusing all the subcomponent information with a Hybrid Bayesian Network (HBN) to provide parameterised output which we use to animate the avatar.

1 Introduction

A human face can exhibit complex and intricate expressions. Facial expression changes are dependent on many factors such as muscle contractions, current emotional state and its implied context. Also facial expressions are individually independent: no two people exhibit the same expression in the same way. These factors make modelling and recognising facial expressions a challenging task.

Bettinger et al. (2002) used AAM (Active Appearance Model) as the underlying basis of their model, sample mean shift and variable length Markov model, to learn the relationships between trajectories of facial expressions, Devin and Hogg (2001) combined AAM with sound as their framework to produce sequences of a talking head. Both approaches do not deal with the expression classification directly. Cohen et al. (2002) used a model based on the motion vectors of Bezier volumes. These vectors were then used in conjunction with a multi-level HMM to classify expression from image sequences. They also experimented with static Bayesian Networks (BN). Chuang et al. (2002) used statistical appearance representation (similar to Cootes and Taylor (2001)) to represent facial expression configurations, then a factorised bilinear model to synthesise existing sequences with different expressions during the speaking process. Tian et al. (2001) used FACS (Facial Action Coding System Ekman et al. (1972)) and a neural network to perform detailed classification of facial expressions. Their approach does not deal with the self occlusion and relies on the detailed geometrical measurements to describe different features which is unreliable.

In this work we wish to model the semantics of a set of low-level facial behaviours, or states which include neutral, smile, grin, surprise, fear, sadness and anger. We aim to model the intrinsic inner-expression relationships by placing hierarchical constraints to bootstrap the process to help in classification of facial expressions. In contrast to Tian et al. (2001); Chuang et al. (2002), we provide one compact and unified probabilistic framework for such a task. Our facial appearance under varying expressions is based on a statistical appearance model originally introduced by Cootes and Taylor (2001). We extend the basic definition of the AAM model to implicitly incorporate parameters for large pose variations into the statistical distribution. Our model is also equipped with a pose estimator to bootstrap the tracking process during large pose changes.

Facial expression classification is achieved by two components: 1) hierarchical shape model, onto which the current instance is projected, where the face is decomposed into the root component consisting of jaw outline, centroids of the eyes and mouth and nose outline. The children are defined as left eye and left eyebrow (eyeL), right eye and right eyebrow (eyeR) and mouth. The children are modelled using PPCA (Section 2.1) and built with frontal view only, letting the pose parameters, such as rotation and translation be inherited from the root component; 2) Hybrid Bayesian Network which fuses all the information to produce the final output. Figure 1 depicts an overview of our system.

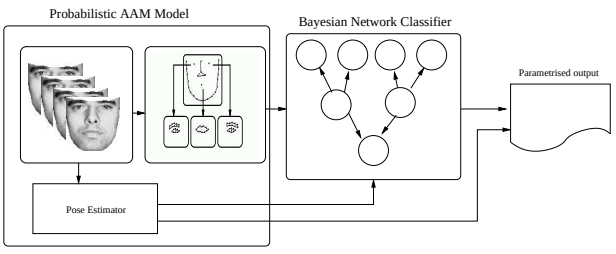


Figure 1: General overview of our system.

2 Framework

The basic representation of an AAM is only able to cope with frontal/near-frontal views ($[-20^o, 20^o]$ in yaw). At the extreme pose changes due to occlusion during the warping process distorts the texture, creating large residuals and causing tracking failure. To overcome this problem we use the pose estimator (Section 2.2) to obtain yaw rotation and mirror the warped image when necessary ($-15^o < yaw > 15^o$). A similar approach was used by Dornaika and Ahlberg (2003). Figure 2 shows the original images from a sequence (top row), frontal view warped texture vectors, with visible distortions (middle row) and pose corrected frontal view texture vectors (bottom row).

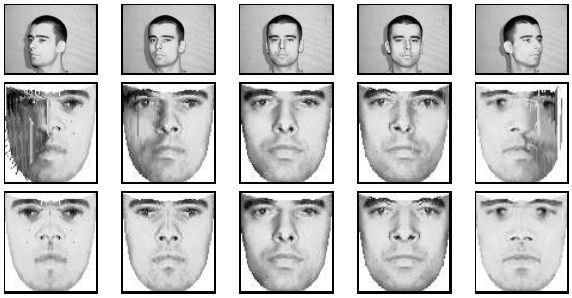


Figure 2: Distortions due to the pose changes and self-occlusion. Top row: original images, middle row: frontal view warped images, bottom row: pose corrected frontal view morphed images.

Unfortunately, mirroring provides only an approximation to the true representation of the face at extreme views. To further improve the tracking process we introduce a pose corrected weight vector such that the original texture difference $\Delta\mathbf{T} = \mathbf{T}_{im} - \mathbf{T}_m$ becomes $\Delta\mathbf{T}_{corr} = \mathbf{T}_w \otimes \Delta\mathbf{T}$, where $\mathbf{T}_{im}$, $\mathbf{T}_m$ are the texture instances in the image and model frame respectively, $\mathbf{T}_w$ is the pose dependent weight vector drawn from the normal distribution and $\otimes$ is component-wise multiplication. Figure 3 shows different representations of $\mathbf{T}_w$ with respect to different yaw rotation values.

Also during the training stage we find the relationship between yaw rotation and the model component responsible for yaw changes by fitting second order polynomial to the data (we are only interested in yaw rotation it is the most likely cause of self-occlusion). During the model fitting stage we use it to provide a model prediction, such that:

$$t_p = a\alpha_h^2 + b\alpha_h + c \tag{1}$$

where α_h is the yaw rotation for the current hypothesis, a, b, c are the coefficients of the polynomial and t_p is the predicted pose parameter such that $t_p \in [-E_h std_{pse}, +E_h std_{pse}]$ with E_h being the residual error of the current hypothesis and std_{pse} being the standard deviation for the given pose component. Experimental results of the pose corrected AAM tracking are presented in Section 4.

Figure 3: Different weight vector representations for different yaw rotation values.

2.1 Probabilistic PCA

PCAs lack of probability distribution makes it-ill suited for the Bayesian framework. Tipping and Bishop (1998) reformulated PCA as the maximum likelihood solution using a latent variable model such that the observed variable $\mathbf{t}$ is given by:

$$\mathbf{t} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \tag{2}$$

where $\mathbf{x}$ is the latent variable such that $P(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{I}_q)$ and $\mathcal{N}$ denotes a Gaussian distribution, $\mathbf{W}$ is the parameter matrix whose columns define the principal subspace of the data, $\boldsymbol{\mu}$ is the d-dimensional vector, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_d)$ where σ^2 is the noise variance, $\mathbf{I}$ is the identity matrix and $\mathcal{N}$ represents a Gaussian distribution. Then

$$P(\mathbf{t}|\mathbf{x}) = \mathcal{N}(\mathbf{t}|\mathbf{W}\mathbf{x} + \boldsymbol{\mu}, \sigma^2\mathbf{I}_d) \tag{3}$$

Marginal distribution of the observed variable $\mathbf{t}$ is

$$P(\mathbf{t}) = \int P(\mathbf{t}|\mathbf{x})P(\mathbf{x})d\mathbf{x} = \mathcal{N}(\boldsymbol{\mu}, \mathbf{C}) \tag{4}$$

where covariance matrix $\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}_d$. The above model represents a constrained Gaussian distribution controlled by $\boldsymbol{\mu}, \mathbf{W}$ and σ^2. A maximum likelihood solution for the parameters is given by:

$$\boldsymbol{\mu}_{ML} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{t}_i \tag{5}$$

$$\mathbf{W}_{ML} = \mathbf{U}_q(\boldsymbol{\Lambda}_q - \sigma^2\mathbf{I}_q)^{\frac{1}{2}}\mathbf{R} \tag{6}$$

$$\sigma^2_{ML} = \frac{1}{d-q}\sum_{i=q+1}^{d}\lambda_i \tag{7}$$

where $\mathbf{t}_i$ is the i-th d-dimensional feature vector from the data set, $\boldsymbol{\Lambda}_q$ is the diagonal matrix containing the q-largest eigenvalues λ_i, $\mathbf{U}_q$ is the matrix containing the q-largest eigenvectors and $\mathbf{R}$ is an arbitrary orthogonal rotation matrix.

2.2 Pose Estimator using PPCA

The pose estimator provides us with continuous 3D pose estimation based on a probabilistic framework. We use a sparse set of training samples, that cover only part of the view sphere, $(-40^o, 40^o)$ around yaw and $(-20^o, 20^o)$ around pitch (using 10^o intervals) and are able to estimate the pose for a much larger, continuous view sphere. The model is also able to generalise to a much denser shape model for appearance synthesis at virtual views. Additionally we do not need to utilise any temporal information such that the estimation is done on-the-fly frame-wise in real time and the system is able to cope with very large jumps and discontinuities in pose change.

Given a d-dimensional multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C}$ its marginal q-dimensional marginal multivariate distribution (where $q << d$) is also Gaussian (Krzanowski, 1988). Let $\mathbf{B}$ be a qxd dimensional identity matrix ($\mathbf{B} = \mathbf{I}$). Then the marginal q-component multivariate probability distribution function (p.d.f) f_q is given by:

$$f_q \sim \mathcal{N}(\mathbf{B}\boldsymbol{\mu}, \mathbf{B}\mathbf{C}\mathbf{B}^T) \tag{8}$$

Following the concept of the marginal p.d.f we define the cumulative distribution function (c.d.f) Φ, such that for a q-dimensional random variable $\mathbf{x}$ the Gaussian c.d.f is given by:

$$\Phi(\mathbf{x}) = \int_{-\infty}^{\mathbf{x}} f_q(\mathbf{x}) d\mathbf{x} \tag{9}$$

We are mostly interested in the c.d.fs that are closely related to the components responsible for the yaw and pitch rotation. Let f_{my}, f_{mx} be marginal p.d.fs and Φ_{my} and Φ_{mx} be marginal c.d.fs corresponding to pose changes. For a given shape $\mathbf{t}$ the estimate of the yaw rotation r_y is given by Equation (10), where a_1, a_2, a_3, a_4 are coefficients of a cubic polynomial estimated during the training stage, p_y is the marginal cdf for the yaw rotation and ϵ_y is the error term defined by constant weighted by the marginal probability f_{my}:

$$\begin{aligned} r_y &= a_1 * p_y^3 + a_2 * p_y^2 + a_3 * p_y + a_4 + \epsilon_y \\ p_y &= \Phi_{my}(\mathbf{t}) \\ \epsilon_y &= f_{my}(\mathbf{t}) * const \end{aligned} \tag{10}$$

The estimate of the pitch rotation r_x is given by Equation (11) where b_1, b_2, b_3, b_4 are coefficients of a cubic polynomial estimated during the training stage, p_x is the marginal cdf of the pitch rotation and ϵ_x is the error term defined by constant weighted by the marginal probability f_{mx}:

$$\begin{aligned} r_x &= b_1 * p_x^3 + b_2 * p_x^2 + b_3 * p_x + b_4 + \epsilon_x \\ p_x &= \Phi_{mx}(\mathbf{t}) \\ \epsilon_x &= f_{mx}(\mathbf{t}) * const \end{aligned} \tag{11}$$

To find the relationship between the angles and the c.d.fs, we use the posterior distribution of the PPCA model. PPCA is described in Section 2.1. We have found that such a probabilistic framework provides much more accurate estimation than one using conventional PCA (e.g. by finding the relationship between the projected parameters and angles). Our model is able to generalise to a denser shape model (Figure 4). This is achieved by down-

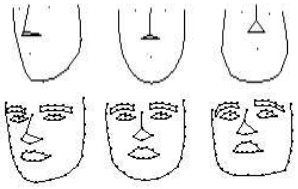


Figure 4: Denser shape model (74 landmarks) driven by the sparse set using 14 landmark points.

sampling the larger PDM to the required size by calculating the centroids of the eyes and mouth and selecting the subset of the jaw outline.

2.3 Hierarchical Shape Representation

We define a hierarchical decomposition of the shape as follows: The jaw outline, nose and centres of the eyes and mouth form the root of our hierarchy. As leaves, or children, we have eye and eyebrow pairs and mouth. Figure 5 shows an example of such decomposition.

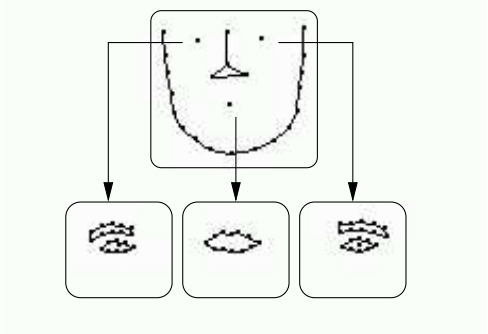


Figure 5: The top row corresponds to the highest point in the hierarchy (root), the middle row corresponds to the leaves.

We allow pose parameters to be incorporated into the root model, and let the children to be frontal view only, with the pose parameters (rotation, translation) inherited from the parent. If the instance of the model at the current frame j has the rotation parameters given by $\mathbf{p}_j = (\alpha, \beta, \gamma)^T$ representing yaw, pitch and bank rotation respectively, the frontal representation $\mathbf{F}_j^i$ for a given instance $\mathbf{M}_j^i$ where $i \in \{eyeL, eyeR, mouth\}$ is then given by:

$$\mathbf{F}_j^i = \mathbf{R}(\mathbf{p}_j)(\mathbf{M}_j^i - \mathbf{T}^i) \tag{12}$$

where $\mathbf{T}^i$ is the translation obtained from the root for i-th leaf. Thus our face can be represented as a combination of the subcomponents. Given the instances of the hierarchical subcomponents, any arbitrary view/expression can be represented as:

$$\mathbf{x}_{final} = B(\mathbf{x}_{root}, \mathbf{x}_{eyeR}, \mathbf{x}_{eyeL}, \mathbf{x}_{mouth}) \quad (13)$$

where B is a shape blending function, $\mathbf{x}_{root}$ is the root model and $\mathbf{x}_{eyeR}, \mathbf{x}_{eyeL}, \mathbf{x}_{mouth}$ are right eye, left eye and mouth models respectively. Our motivation for choosing hierarchical representation of the shape model is as follows: We strongly believe that the shape is individually independent (given appropriate normalisation) and can be efficiently utilised to capture manifolds of the facial expressions.

We also noticed that the modes of variation for each of the components correspond to their intrinsic functionality. For example for mouth they are mouth open, mouth closed and mouth grin. We associate corresponding marginal and cumulative probability distributions with each of the functionalities. Instead of defining facial expressions as holistic entities, we represent them as a combination of intrinsic functionalities of the subcomponents (expression implied facial feature independence has been exploited by Donato et al. (1999); Zalewski and Gong (2004)). So any facial expression can be defined as:

$$expression = state_{mouth} + state_{eyeR} + state_{eyeL}$$

The advantages of this are two-fold: First of all, each of the expressions is defined in a more intuitive and quantitative way. Secondly, such a representation allows us to account for similar expressions (smile with eyes open, or smile with eyes closed) without any additional overhead. Given probability distributions for each of the subcomponents, we obtain final classification by fusing all the information through a Hybrid Bayesian Network (Section 3), hence producing a parameterised form of expression definition.

3 Expression Classification

Bayesian Networks allow us a way for data fusion in a probabilistic fashion, and have been successfully used in face recognition and classification related tasks (Yand et al., 2002). As the basis of the classifier, we adopt a Hybrid Bayesian Network (HBN) (Figure 6).

Round nodes correspond to continuous states, square ones to the discrete states. Shaded nodes are observed, unshaded ones are hidden. In the design of this HBN we took into consideration psychophysical evidence implied by the human perception of facial expressions (Ekman, 1973; Ekman et al., 1972). Such a layout gives us the means to describe the states of each of the subcomponents at a high abstraction level that in turn can be used as a parameterised output for animation purposes.

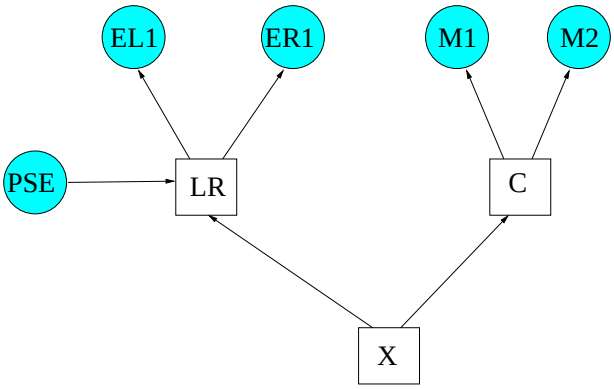


Figure 6: The Hybrid Bayesian Network used for a paramtrised expression definition.

We derive logical sections within the net that characterise the functionalities of the different facial components. These are eye components defined by likelihoods $P(EL|LR)$, $P(ER|LR)$, and mouth component defined by likelihoods $P(M1|C)$, $P(M2|C)$, which are drawn from our hierarchical model such that

$$\begin{aligned} P(EL|LR) &= f^1_{eyeL}(\mathbf{t}_{eyeL}) \\ P(ER|LR) &= f^1_{eyeR}(\mathbf{t}_{eyeR}) \\ P(M1|C) &= f^1_{mouth}(\mathbf{t}_{mouth}) \\ P(M2|C) &= f^2_{mouth}(\mathbf{t}_{mouth}) \end{aligned}$$

where f^j_i defines the posterior marginal probability distribution for the j-th principal component, $i \in \{eyeL, eyeR, mouth\}$ and $\mathbf{t}_i$ is the input vector. The prior $P(PSE)$ is drawn from the marginal distribution of the pose model such that:

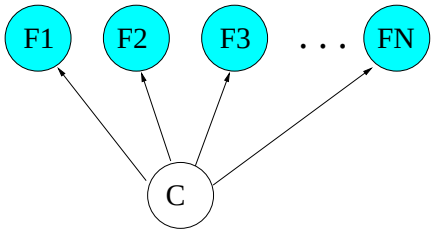
$$P(PSE) = f^1_{pose}(\mathbf{t}_{pose})$$

and accounts for the missing features in extreme pose changes. If one of the features becomes occluded, the remaining visible feature, not the combination of both, will be used for classification.

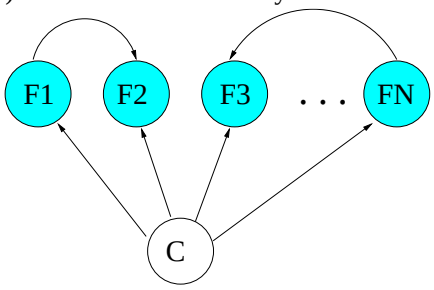
We can think of the output nodes as descriptors of different features on the face, which are independent of each other (this is implied by orthogonality of our distribution spaces). As Cohen et al. (2002) pointed out, the main limitation of feature based Naive Bayesian Classifiers is the independence of the features given the expression which might not be true in real life scenarios. (Figure 7 (a)) depicts such a Naive Bayesian Classifier, where $F1, F2, F3, ..., FN$ define different features and C is the expression class. To overcome that limitation they suggested use of a TAN classifier (Tree Augmented Naive) where dependencies are represented as arcs between different features, and its structure in defined is the learning stage (Figure 7 (b)).

In our case, to account for dependencies amongst different facial features we introduce hidden nodes LR and

C. During training these nodes will capture the possible dependencies among different feature inputs (LR for EL, ER and C for $M1, M2$).



(a) Feature-based Naive Bayesian Classifier



(b) Feature-based TAN Classifier

Figure 7: Different representations of BNs.

To perform final classification we choose the hypothesis that maximises the posterior given the evidence Θ and the net structure $\mathbf{m}$:

$$P(\mathbf{m}|\Theta) = P(X|LR, C, A, B, EL, ER, M1, M2, PSE, \Theta) \propto P(X, LR, C, A, B, EL, ER, M1, M2, PSE|\Theta)$$

To train the Hybrid Bayesian Net we performed supervised training based on 813 hand labelled samples representing continuous sequences of various changes in facial expressions.

4 Experiment

For AAM model training we used a set consisting of 1790 images and shapes (74 landmarks), which included seven basic expressions (neutral, smile, grin, sadness, fear, anger, surprise) and large variations in pose. For the frontal view, a hierarchical decomposition shape model, a training set of 700 shapes was used. For the pose estimator, 640 different and much sparser shapes (14 landmarks) were used. All the training samples were hand labelled beforehand.

The outline of our algorithm is as follows:

- Perform colour segmentation on the input image to find a rough position of the head and remove unnecessary background information. For colour segmentation, the HSV (Hue,Saturation,Value) colour model is used, which carry sufficient discriminative information for such a task.
- Fit the pose-corrected AAM model representation to the image. Obtain a pose estimate through from the shape model (repeat both until convergence).
- Obtain final pose estimate.
- Project the current instance of the model onto the hierarchical shape model.
- Classify the expression using the Hybrid Bayesian Network and obtain a parameterised output.
- Animate Avatar according to the obtained parameterised output.

The Avatar animation was performed using a morph-based approach (Noh and Neumann, 1998), defined by:

$$Expression = \sum_i \mathbf{w}(i)\mathbf{\Gamma}(i) \qquad (14)$$

where $\mathbf{w}$ defines the morph weight vector such that $\sum_i \mathbf{w}(i) = 1$ and $\mathbf{\Gamma}$ defines a set of morph bases. To test our improved AAM representation we used a test sequence 0 containing 415 frames and large pose changes. Figure 8 shows the first few frames. Top row corresponds to the original AAM formulation, with visible loss of focus at large pose variation, and the bottom row corresponds to the pose corrected AAM, where the focus is not greatly affected by pose changes.

To test the expression classifier we used two sequences containing 750 and 530 frames respectively. We compared this with the BN given in the Figure 7 (3 input nodes, taking the parameter vectors from the hierarchical distribution). We obtained the following classification rates:

	Our HBN	Cohen et al. (2002)
test sequence 1	88%	82%
test sequence 2	83%	80%

Figure 11 shows selected frames from the sequence 1 experiment. Within each of the boxes the left image corresponds to the currently tracked image frame with the AAM mask superimposed on it. The image on the right corresponds to the synthetic avatar animated according to the classified expression. Figure 9 shows the corresponding classification results for test sequence 1 and Figure 10 for test sequence 2.

5 Conclusion

In this paper we have extended the basic AAM approach to cope with large pose variations by introducing pose-based constraints upon the tracking process. We also introduced shape based hierarchical decomposition of a human face into independent components, such that their

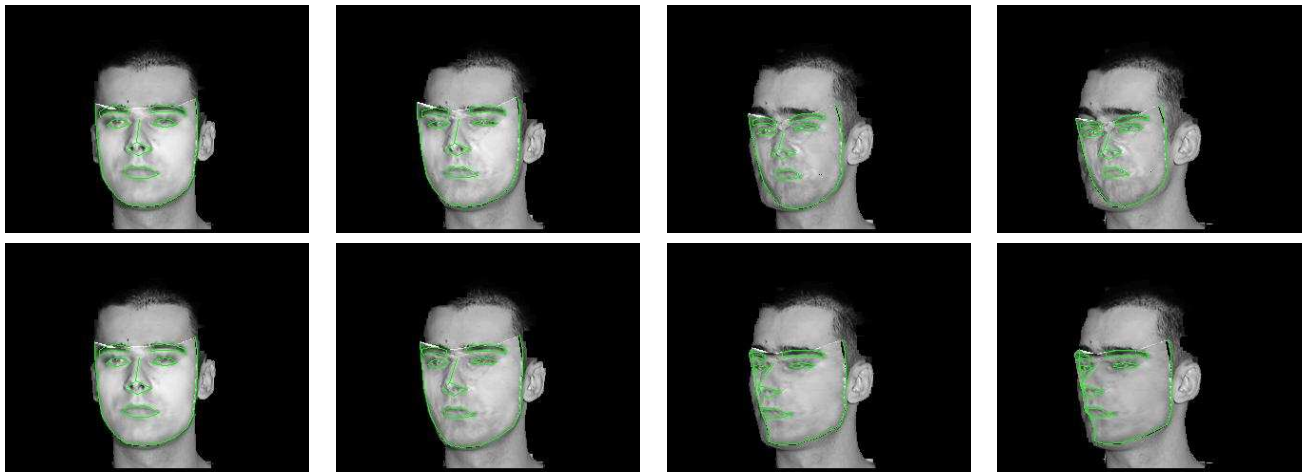


Figure 8: Selected frames from the experiment on the AAM fitting onto the extreme pose view. Top row corresponds to the original AAM formulation and bottom row to the pose corrected AAM.

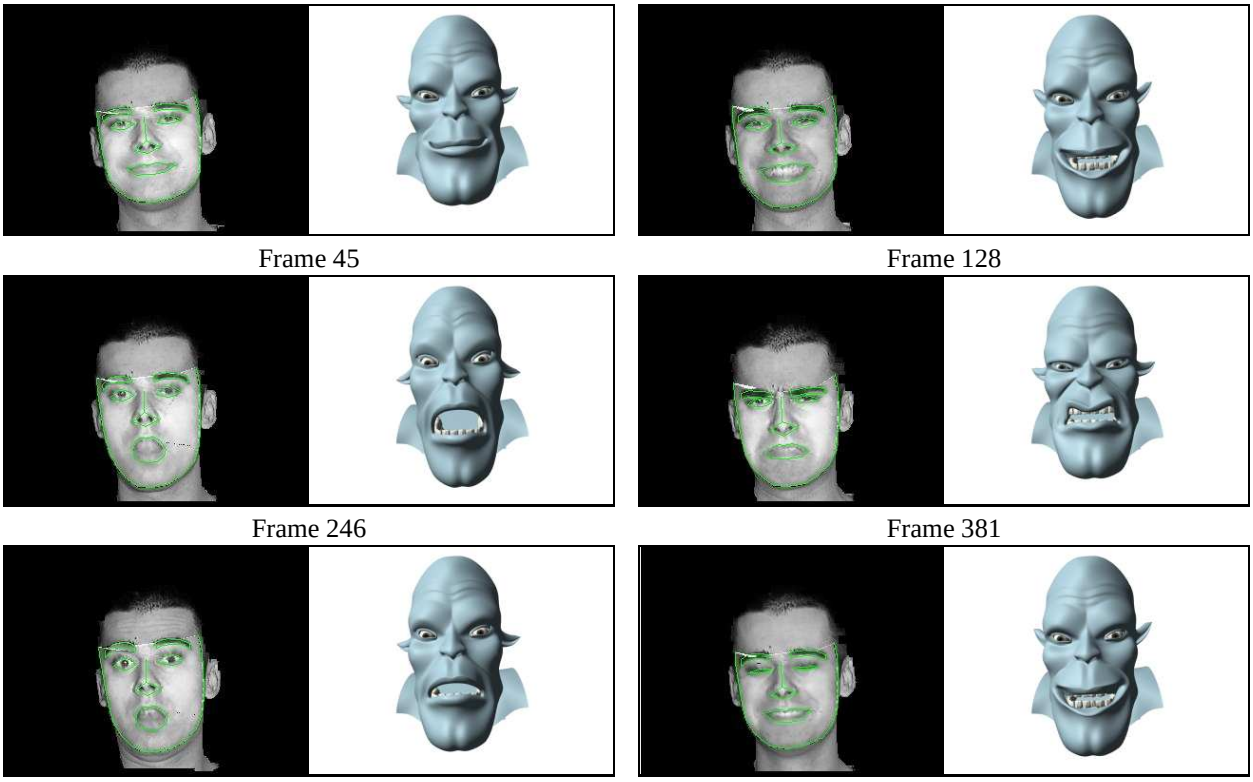


Figure 11: Selected frames from the experiment on expression classification and avatar animation (test sequence 1). Each of the images shows tracked frame with AAM mask superimposed on it (left) and corresponding synthesised avatar (right).

combinatorial form can be used to define an arbitrary facial expression, and the probabilities obtained from their distributions in conjunction with Hybrid Bayesian Network (HBN) can serve as a basis for expression classification. Our future work includes investigation into Dynamic Bayesian Networks (DBNs) and their use in behaviour context modelling.

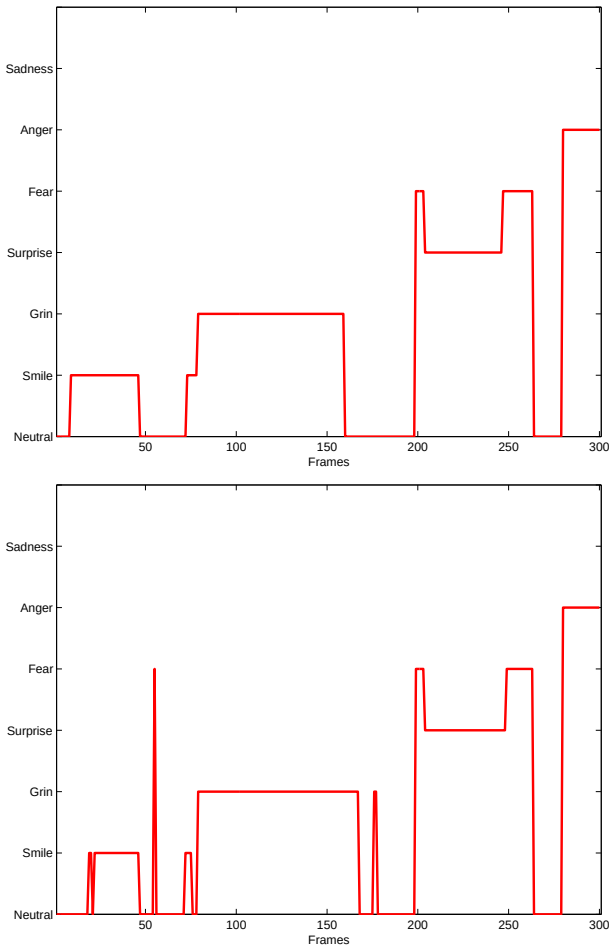


Figure 9: Expression classification for test sequence 1: top row corresponds to our HBN approach, bottom row to the Cohen et al. (2002) approach.

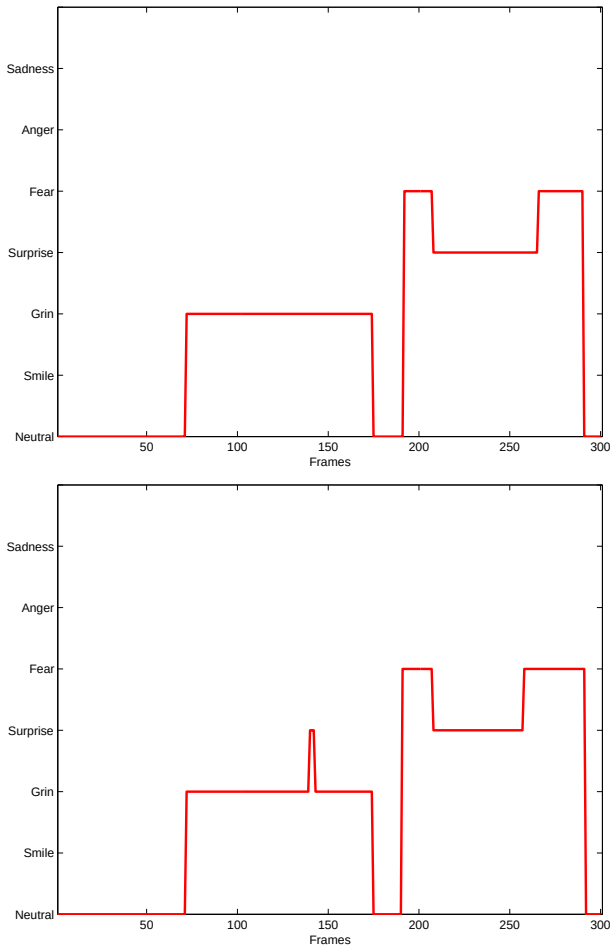


Figure 10: Expression classification for test sequence 2: top row corresponds to our HBN approach, bottom row to the Cohen et al. (2002) approach.

Acknowledgements

We would like to thank Jose Galan for enlightening discussions on Bayesian Networks.

References

F. Bettinger, T. F. Cootes, and C. J. Taylor. Modelling facial behaviours. In *BMVC*, volume 2, pages 797–806, 2002.

E. S. Chuang, H. Deshpande, and C. Bregler. Facial expression space learning. In *10th Pacific Conference on Computer Graphics and Applications*, Beijing, 2002.

I. Cohen, N. Sebe, L. Chen, A. Garg, and T. S. Huang. Facial expression recognition from video sequences. *International conference on Multimedia and Expo*, 2:121–124, 2002.

T. F. Cootes and C. J. Taylor. Statistical models of apperance for computer vision. Technical report, University of Manchester, Manchester, UK, 2001.

V. E. Devin and D. C. Hogg. Reactive memories: An interactive talking head. In *BMVC*, 2001.

G. Donato, M. S. Barlet, J. C. Hager, P. Ekman, and T. J. Sejnowski. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974–989, October 1999.

F. Dornaika and J. Ahlberg. Effi cient active appearance model for real-time head and facial feature tracking. *2003 IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, pages 173–180, October 2003.

P. Ekman, editor. *Darwin and facial expressions: A century of research in review*. Academic Press New York, 1973.

P. Ekman, W. V. Frieser, and P. Ellsworth. *Emotion in the human face*. Pergamon New York, 1972.

W. J. Krzanowski. *Principles of Multivariate Analysis*. Oxford University Press, 1988.

J. Noh and U. Neumann. A survey of facial modeling and animation techniques. Technical report, University of Southern California, 1998.

Y. Tian, T. Kanade, and J. F. Cohon. Recognising action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):1–19, February 2001.

M. E. Tipping and C. M. Bishop. Mixture of probabilistic component analysers. Technical report, Dept of Computer Science and Applied Mathematics Aston University, Birmingham B4 7ET, UK, 1998.

M. Yand, D. Kriegman, and N. Ahuja. Detecting faces in images: A survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 34–58, 2002.

L Zalewski and S. Gong. Synthesis and recognition of facial expressions in virtual 3d views. In *Proc. 6th IEEE International Conference on Automatic Face and Gesture Recognition*, Korea, 2004.