

FAME – A Flexible Appearance Modelling Environment

Mikkel B. Stegmann, Bjarne K. Ersbøll, and Rasmus Larsen

Mikkel B. Stegmann [mbs@imm.dtu.dk], Bjarne K. Ersbøll [be@imm.dtu.dk] and Rasmus Larsen [rl@imm.dtu.dk] are with Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby.

Abstract

Combined modelling of pixel intensities and shape has proven to be a very robust and widely applicable approach to interpret images. As such the Active Appearance Model (AAM) framework has been applied to a wide variety of problems within medical image analysis. This paper summarises AAM applications within medicine and describes a public domain implementation, namely the Flexible Appearance Modelling Environment (FAME). We give guidelines for the use of this research platform, and show that the optimisation techniques used renders it applicable to interactive medical applications. To increase performance and make models generalise better, we apply parallel analysis to obtain automatic and objective model truncation. Further, two different AAM training methods are compared along with a reference case study carried out on cross-sectional short-axis cardiac magnetic resonance images and face images. Source code and annotated data sets needed to reproduce the results are put in the public domain for further investigation.

Keywords

Active appearance models, face segmentation, left ventricular segmentation, public domain training data and software.

I. INTRODUCTION

Modelling biological variation present in medical images has been a challenge for a long time. The variation is often very complex and each image modality has its own set of characteristic artefacts. While the amount of biological complexity is either fixed or very slowly changing, new imaging modalities emerge and existing are improved, often very rapidly. This accentuates the need for general methods capable of adapting to both issues above. In addition, we would prefer these to be reasonably fast, specific to the current problem and finally, robust to noise and acquisition artefacts.

During the past few years, the generative modelling framework: Active Appearance Models (AAM) [1], [2] have aimed at meeting all of the above requirements. By being capable of synthesising near photo-realistic images of objects, AAMs are taking the analysis-through-synthesis approach to the extreme. This approach has proven its worth in numerous different applications, both medical and non-medical.

We sincerely believe that progress is most easily obtained through *transparency*. By this we mean that algorithms and reference data set should be made available along with corresponding reference performance measures. In that way new and existing methods can be compared on a fair and transparent basis by eliminating, or at least significantly alleviate, the tedious process of re-implementing competing methods and re-generating similar training data, et cetera.

This paper represents a small step towards transparent research by presenting a public domain AAM implementation along with reference measures carried out on likewise public domain data. Focus is put on the different design choices an AAM designer must make prior to model building. In particular, we evaluate two different variations on the AAM formulation and treat the bias/variance problem occurring when selecting the number of principal components.

The paper is organised as follows: Section II summarises medical applications of AAMs. Section III serves as an introduction to the inner workings of AAMs. Section IV describes the two prevalent methods for calculating model parameter updates. Section V treats the problem of selecting the number model

parameters. Section VI describes our AAM framework, The Flexible Appearance Modelling Environment (FAME). Section VII describes two sets of training data and gives segmentation results on these using FAME. Finally, Section VIII serves a discussion and conclusion.

II. BACKGROUND

At the introduction of Active Appearance Models [1], [2] focus was put on segmentation and interpretation of face images. However, being a generic approach to build models from an annotated training set, medical applications were soon to follow. This section summarises the use of AAMs in medical imaging, but the list is by no means exhaustive. For example, the vast amount of literature on AAM-related variations and precursors has been omitted.

Both [3], [4], [5], [6] showed AAMs modelling ventricles, the caudate nucleus and the lentiform nucleus in single-slice magnetic resonance images (MRI) of the brain. This case was also treated in [7] and [8] where the latter also demonstrated an AAM of single-slice MRI of cartilage and bone in the knee. Up to this point medical AAM work was done by the inventors, primarily Cootes, Taylor and Edwards of Manchester University.

Mitchell et al. [9], [10] demonstrated the applicability of AAMs on single-slice cardiac MRI and later on time series of cardiac MRI [11], [12], [13]. Bosch et al. [14], [15] used a similar approach on time series of echocardiograms. A direct continuation of this work is the 3D AAMs built on volumetric cardiac MRI by Mitchell et al. [16], [17] and Lelieveldt et al. [18]. Two variations of 3D AAMs were applied to diaphragm dome CT by Beichel et al. [19], [20] and to echocardiogram time series by Bosch et al. [21]. Duchesne et al. [22] have used a variation of the traditional AAM, with a shape model similar to that of the Morphable Models of Jones and Poggio [23], for 3D segmentation of the hippocampus in brain MRI. Stegmann et al. [24], [25], [26], [27] used AAMs for segmentation of metacarpal radiographs and cardiac MRI. Later Thodberg [28] used a Shape AAM for segmentation of metacarpal radiographs. A modified AAM, using multiple texture models, was used for segmenting myocardial perfusion MRI sequences by Stegmann and Larsson [29], [30]. Hilger et al. [31] explored a noise robust texture representation on ventricular cardiac MRI. Finally, Stegmann et al. [32], [33] used sequential AAMs based on MDL-defined landmarks for corpus callosum analysis in brain MRI.

III. ACTIVE APPEARANCE MODELS

Active Appearance Models establish a compact parameterisation of object variability, as learned from a training set by estimating a set of latent variables. The modelled object properties are usually shape and pixel intensities. The latter is henceforward denoted *texture*. From these quantities new images similar to the training set can be synthesised.

Objects are defined by marking up each example with points of correspondence (i.e. landmarks) over the set either by hand, or by semi- to completely automated methods. From these landmarks a shape

model [34] is built. Further, given a suitable warp function a dense (i.e. per-pixel) correspondence is established between training objects, thereby enabling a proper modelling of texture variability. By exploiting prior knowledge of the nature of the optimisation space, these models of shape and texture can be rapidly fitted to unseen images, thus providing image interpretation through synthesis.

Variability is modelled by means of a Principal Component Analysis (PCA), i.e. an eigen analysis of the dispersions of shape and texture. Let there be given P training examples for an object class, and let each example be represented by a set of N landmark points and M texture samples. The shape examples are aligned to a common mean using a Generalised Procrustes Analysis (GPA) [35] where all effects of translation, rotation and scaling are removed. The obtained Procrustes shape coordinates are subsequently projected into the tangent plane to the shape manifold, at the pole given by the mean shape. The texture examples are warped into correspondence using a piece-wise affine warp and subsequently sampled from this *shape-free* reference. Typically, this geometrical reference frame is the Procrustes mean shape. Let \mathbf{s} and \mathbf{t} denote a synthesized shape and texture and let $\bar{\mathbf{s}}$ and $\bar{\mathbf{t}}$ denote the corresponding sample means. New instances are now generated by adjusting the PC scores, \mathbf{b}_s and \mathbf{b}_t in

$$\mathbf{s} = \bar{\mathbf{s}} + \Phi_s \mathbf{b}_s \quad , \quad \mathbf{t} = \bar{\mathbf{t}} + \Phi_t \mathbf{b}_t \quad (1)$$

where Φ_s and Φ_t are matrices of column eigenvectors of the shape and texture dispersions estimated from the training set. To obtain a combined shape and texture parameterisation, \mathbf{c} , the values of \mathbf{b}_s and \mathbf{b}_t over the training set are combined into

$$\mathbf{b} = \begin{bmatrix} \mathbf{W}_s \mathbf{b}_s \\ \mathbf{b}_t \end{bmatrix} = \begin{bmatrix} \mathbf{W}_s \Phi_s^T (\mathbf{s} - \bar{\mathbf{s}}) \\ \Phi_t^T (\mathbf{t} - \bar{\mathbf{t}}) \end{bmatrix}. \quad (2)$$

A suitable weighting between pixel distances and pixel intensities is carried out through the diagonal matrix \mathbf{W}_s . To make the normalised measures of pixel distance and pixel intensities commensurate, the shape PC scores are typically weighted by the square root of the ratio between the sums of the texture and shape eigenvalues.

To recover any correlation between shape and texture the two eigenspaces are usually coupled through a third PC transform

$$\mathbf{b} = \Phi_c \mathbf{c} = \begin{bmatrix} \Phi_{c,s} \\ \Phi_{c,t} \end{bmatrix} \mathbf{c} \quad (3)$$

obtaining the combined appearance model parameters, \mathbf{c} , that generates new object instances by

$$\mathbf{s} = \bar{\mathbf{s}} + \Phi_s \mathbf{W}_s^{-1} \Phi_{c,s} \mathbf{c} \quad , \quad \mathbf{t} = \bar{\mathbf{t}} + \Phi_t \Phi_{c,t} \mathbf{c}. \quad (4)$$

To regularise the model and improve speed and compactness, Φ_s , Φ_t and Φ_c are truncated, usually such that a certain amount of variance in the training set is preserved. This eventually results in k ($k < P$) combined modes, i.e. k dynamic parameters encoded in the vector \mathbf{c} .

The object instance, (\mathbf{s}, \mathbf{t}) , is synthesised into an image by warping the pixel intensities of \mathbf{t} into the geometry of the shape \mathbf{s} . Given a suitable similarity measure the model is matched to an unseen image using an iterative updating scheme based on a fixed Jacobian estimate [36], [8] or a principal component regression [2]. Typically the L_2 -norm is used as similarity measure, which is sensible if the errors between the model and the image are (approximately) normally distributed.

This sums up the basic theory of AAMs. For further details refer to [2], [36], [8] (and [24] as a supplement).

IV. AAM TRAINING

Traditionally, AAMs have been trained to update model and pose parameters using one of two schemes described in the following. These parameter updates are carried out using difference images between the current model image and the corresponding part of the unseen image that it covers. Applying such parameter corrections in an iterative scheme should drive the model towards the ground truth shape in the image.

Multivariate Regression

The original AAM formulation [1], [2] uses a regression approach where difference vectors, $\delta\mathbf{t} = \mathbf{t}_{image} - \mathbf{t}_{model}$, are regressed onto corresponding parameter perturbation/displacement vectors, $\delta\mathbf{p}$. Here \mathbf{p} is model and/or pose parameters, having the length Q . The goal is thus to obtain an optimal – in a least-squares sense – prediction matrix, \mathbf{R} , satisfying the linear relationship:

$$\delta\mathbf{p} = \mathbf{R}\delta\mathbf{t}. \quad (5)$$

Our formulation uses separate prediction matrices for model and pose parameters, denoted \mathbf{R}_c and \mathbf{R}_ρ , respectively. Let there be conducted S perturbation experiments and let

$$\mathbf{P} = \begin{bmatrix} \vdots & & \vdots \\ \delta\mathbf{p}_1 & \dots & \delta\mathbf{p}_S \\ \vdots & & \vdots \end{bmatrix} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} \vdots & & \vdots \\ \delta\mathbf{t}_1 & \dots & \delta\mathbf{t}_S \\ \vdots & & \vdots \end{bmatrix}. \quad (6)$$

During these perturbation experiments $\delta\mathbf{t}_i$ is short for $\delta\mathbf{t}_i(\mathbf{p}^+ + \delta\mathbf{p}_i)$, where \mathbf{p}^+ denotes the optimal parameter configuration. Since $Q \ll S \ll M$ typically, \mathbf{R} is estimated using principal component regression [8]. From the large matrix \mathbf{T} , an s -dimensional subspace ($s \leq S$) is extracted. This makes \mathbf{R} well-determined in $\mathbf{P} = \mathbf{R}\mathbf{T}'$, where \mathbf{T}' contains s -dimensional projected versions of \mathbf{T} . Consequently, an eigen-decomposition of an $S \times S$ matrix is involved. To keep this matrix down to a feasible size, typically only a subset of the training set is used for training. This is especially important when estimating \mathbf{R}_c , as the number of model modes grows with the number of examples, P . For models with a small number of training examples the growth in S becomes close to quadratic. In the experiments below the same training subset has been used to estimate both \mathbf{R}_c and \mathbf{R}_t .

In the subsequent sections, this learning approach is denoted *Regression*.

The Fixed Jacobian Matrix Estimate

In later AAM publications [36], [7], [37] the multivariate regression is superseded by a simpler approach. It is easier to implement, faster to calculate and requires far less memory to execute. First we introduce the residual vector \mathbf{r} , parameterised by \mathbf{p} :

$$\mathbf{r}(\mathbf{p}) = \delta\mathbf{t}(\mathbf{p}) = \mathbf{t}_{image}(\mathbf{p}) - \mathbf{t}_{model}(\mathbf{p}). \quad (7)$$

A first order Taylor expansion of \mathbf{r} at \mathbf{p}^* is

$$\mathbf{r}(\mathbf{p}^* + \delta\mathbf{p}) \approx \mathbf{r}(\mathbf{p}^*) + \frac{\partial\mathbf{r}(\mathbf{p}^*)}{\partial\mathbf{p}}\delta\mathbf{p} \quad (8)$$

where \mathbf{p}^* is in the proximity of \mathbf{p}^+ and

$$\frac{\partial\mathbf{r}(\mathbf{p}^*)}{\partial\mathbf{p}} = \frac{\partial\mathbf{r}}{\partial\mathbf{p}} = \begin{bmatrix} \frac{\partial r_1}{\partial p_1} & & \frac{\partial r_1}{\partial p_Q} \\ \vdots & \dots & \vdots \\ \frac{\partial r_M}{\partial p_1} & & \frac{\partial r_M}{\partial p_Q} \end{bmatrix}. \quad (9)$$

The goal of the parameter update is to drive the residual vector to zero, i.e. finding \mathbf{p}^+ . Using the L_2 -norm the optimal $\delta\mathbf{p}$ is: $\arg \min_{\delta\mathbf{p}} \|\mathbf{r}(\mathbf{p}^* + \delta\mathbf{p})\|^2$. Hence, the least-squares solution of (8) becomes:

$$\delta\mathbf{p} = - \left(\frac{\partial\mathbf{r}}{\partial\mathbf{p}}^\top \frac{\partial\mathbf{r}}{\partial\mathbf{p}} \right)^{-1} \frac{\partial\mathbf{r}}{\partial\mathbf{p}}^\top \mathbf{r}(\mathbf{p}^*) = -\mathbf{R}\mathbf{r}(\mathbf{p}^*). \quad (10)$$

To obtain optimal numerical stability, a singular value decomposition (SVD) of the Jacobian matrix, $\frac{\partial\mathbf{r}}{\partial\mathbf{p}}$ is preferred in order to obtain its pseudo-inverse, \mathbf{R} . However due to the size this is not feasible, which is why a normal matrix inversion must be carried out.

Normally, the Jacobian matrix must be recalculated at each \mathbf{p}^* , which is a very expensive task due to its size. However, since AAMs operate in a standardised domain, i.e. the shape-free reference frame, AAMs perform the following approximation

$$\frac{\partial\mathbf{r}(\mathbf{p}^*)}{\partial\mathbf{p}} \approx \frac{\partial\mathbf{r}(\mathbf{p}^+)}{\partial\mathbf{p}}. \quad (11)$$

Further – and this is a somewhat crude assumption – the right-hand side of (11) is considered constant for all training examples. Thus, \mathbf{R} is considered fixed and estimated once during the model building process using numeric differentiation on the training set [36]. Let δp_{jk} denote the k -th perturbation of the j -th parameter, \mathbf{e}_j , a unit-vector for the j -th dimension and $w(\cdot)$, a weighting kernel, e.g. Gaussian [36]. Using a central estimator on all P training examples, the j -th column of the Jacobian is estimated as

$$\widehat{\frac{\partial\mathbf{r}}{\partial p_j}} = \frac{1}{P} \sum_i^P \sum_k \frac{w(\delta p_{jk}) \mathbf{r}(\mathbf{p}_i^+ + \delta p_{jk} \mathbf{e}_j) - \mathbf{r}(\mathbf{p}_i^+ - \delta p_{jk} \mathbf{e}_j)}{2\delta p_{jk}}. \quad (12)$$

In our experiments, the Jacobian matrix is estimated without weighting, i.e. using a uniform kernel, $w(\cdot)$.

In the subsequent sections, this learning approach is denoted the *Jacobian*.

Perturbation Scheme

The remaining design choice in both learning methods is the perturbation scheme. Although being a rather important step in crafting a good AAM, this topic is often not described in AAM literature.

For all experiments shown in this paper we used the perturbation scheme shown in Table I. This has earlier been used over a wide range of cases with good results, see e.g. [29], [30], [31], [33]. However, the perturbation scheme remains a free parameter and could be subject for a thorough study on a per case basis to obtain optimal segmentation accuracy. To compare the two AAM training methods, given one perturbation scheme, each parameter is displaced separately, while the remaining parameters are kept at zero. Alternatively, as mentioned in [38], one could seek the optimal perturbation scheme given a specific training method. See the recent work in [38] for a variation of the AAM training, which allows simultaneous perturbation of all model parameters using both the regression and Jacobian training method.

TABLE I
PERTURBATION SCHEME USED IN BOTH LEARNING METHODS

Variable	Perturbations
x, y	$\pm 5\%$, $\pm 10\%$ of the width and height, respectively
θ	± 5 , ± 15 degrees
$scale$	$\pm 5\%$, $\pm 15\%$
$c_1 \dots c_k$	± 0.25 , ± 0.50 standard deviations

V. MODEL TRUNCATION

Typically, the eigenspaces of shape, texture and combined variation are truncated so that each explains a fixed amount of variance. Since the variance along the i^{th} principal axis is equal to the corresponding eigenvalue, λ_i , this is easily carried out. To retain p percent of the variation, t modes can be chosen satisfying:

$$\sum_{i=1}^t \lambda_i \geq \frac{p}{100} \sum \lambda_i \quad (13)$$

How to estimate p is a classic bias/variance problem. Choosing a high value for p could result in a model too flexible, in the sense that it will be able to fit to noise present in the training set (high variance) and thus not generalise very well. Conversely, low p values would produce very restrictive models, not matching the training set very well (high bias towards the model). A common heuristic is to consider the remaining 5% of the signal to be noise.

An alternative method for choosing p is cross-validation, i.e. partitioning of the data set into a training and an independent test set. By adding modes the reconstruction error will initially decrease. But at some p the model will start to over-fit the training set and the reconstruction error on the test set will start to

increase. To make this procedure less negatively biased and more robust several partitions can be used going towards a leave-one-out analysis in the extreme. More elaborate methods such as the bootstrap [39] could also be applied.

A convenient alternative to cross-validation is parallel analysis introduced by Horn [40], where the data is compared to either i) independent normally distributed synthetic samples, or ii) a randomised version of the data matrix.¹ We will concentrate on the latter since it is imposing less strict distributional assumptions on the data. Further, it is simpler to implement and calculate.

In short, parallel analysis seeks the amount of inflation in the eigenvalues due to sampling errors (also known as *chance capitalisation* or *chance correlation*). In the perturbation version of parallel analysis this is estimated by breaking the correlation between variables, i.e. for each variable the order of the observations is randomised. In the case of a shape model, this would be a randomisation of each landmark over all shapes. From a scree plot of the eigenvalues of the original data and the scrambled data, modes that have higher eigenvalues for the randomised data than the original data, can thus be considered noise. The rationale is that these modes are only stemming from chance capitalisation and not actual population correlation.

This data permutation version of Horn’s parallel analysis is typically embedded in a Monte Carlo simulation scheme, where mean eigenvalues of several scrambling experiments are used to reduce sampling errors. Experiments are done with replacement due to the massive effort involved in keeping track of the permutations.

VI. FAME

The Flexible Appearance Modelling Environment (FAME) is a software framework containing an open source implementation of the Active Appearance Models. This core part of FAME is called the AAM-API. In addition, the FAME package provide *AAMLab* a GUI tool for image annotation, model exploration and model fitting, i/o scripts for communication with Matlab, test data sets etc. The complete package including data material, source code, documentation, scripts etc. needed to reproduce the results below are placed in the public domain and can be downloaded from:

<http://www.imm.dtu.dk/~aam/>

FAME is implemented in C++ and runs under Microsoft Windows. Two third-party libraries, VisSDK [41] and LAPACK [42] are required for image handling and eigen decompositions. Both of these are also placed in the public domain.

Emphasis has been put on high performance and on providing a cornucopia of documentation features. This paper will present several examples of the latter. Due to the unsupervised analyses in AAMs, we believe that a partly exploratory approach to AAM building is required to craft good models.

¹The original work of John L. Horn used the former method.

Details of the implementation are placed in the appendix to provide a quick overview. Appendix A provides observations on and changes from the original AAM formulation. Appendix B enumerates the different design choices prior to model building using FAME.

AAMs rely heavily on image warps. It is often believed that image sampling and warping constitute the major part of an optimisation iteration. We have found that this is not the case. In the current implementation, warps can be carried out in optimised software code or by exploiting available OpenGL compliant graphics hardware. The latter is described in Appendix C. To assess performance and the general applicability of the framework details on machinery and timings are given in Appendix D.

Where the results below are not obtained directly from FAME, simple Matlab scripts were used. All of these had no parameters. Hence, this should not limit the reader from reproducing the results.

A. Extending FAME

FAME includes a command line interface called *aamc* that demonstrates basic usage. All experiments in this article were carried out using this. However, FAME users are encouraged to modify, extend or embed the framework in any way supporting education or research. We use FAME in two ways, i) from the "outside" by calling methods etc. and ii) from the "inside" by deriving new versions of the concepts using C++ inheritance.

VII. CASE STUDIES

This section comprises two cross-sectional case studies, where the first is an example of very complex biological variation in both shape and appearance. Moreover, it has the advantage that all readers have a visual system fine-tuned towards this case study. It consists of a cohort of face images. Examples of medical use of statistical face models include the work of Hutton et al. [43], which is aiming at a better understanding of syndromic facial dysmorphologies, here studying the Noonan syndrome.

The second case study addresses an invariable prerequisite for quantitative functional analysis of the human heart [44], namely segmentation of cardiac chambers. In this study the left and right ventricles are segmented in a set of single-slice cardiac MRI.

A. Face Images

The data set consists of 37 still images of 37 different frontal human faces, all without glasses and with a neutral expression. The gender distribution is 7 females and 30 males. Images were acquired in 640×480 JPEG colour format using a DV video camera. The following facial structures were manually annotated using 58 landmarks in total: eyebrows, eyes, nose, mouth and jaw. A total of seven point paths were used; three closed and four open. Refer to Figure 1 for an example annotation.



Fig. 1. Example annotation of a face using 58 landmarks.

A.1 Shape Model

The foundation of the shape model is the 58 facial landmarks shown in Figure 2 (left). To establish a reference coordinate system relative to these landmarks within their convex hull a Delaunay triangulation is calculated and shown in Figure 2 (right). This constitutes the coordinate system of the texture model.

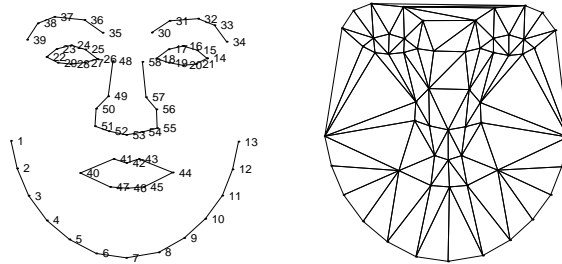


Fig. 2. Facial landmarks (left). Triangular model mesh (right).

Plotting the scatter of all 37 face annotations yields the rather confusing plot in Figure 3 (left). To obtain only shape variation all Euclidean transformations have been filtered out in Figure 3 (right) by means of a Generalised Procrustes Analysis (GPA). Further, the Procrustes mean shape is fully drawn.

A better impression of landmark variation is given in Figure 4 (left), where the principal directions of each point are plotted. This reveals that the point variation of the aligned set of faces is often heteroscedastic, especially for landmarks at the upper jaw and on the lips.

To obtain an impression of landmark correlation, refer to Figure 4 (right) showing the canonical correlation² between each landmark. The five major block diagonals are jaw, eyes, eyebrows, mouth and nose.

Showing the three major modes of shape variation, Figure 5 displays the three directions with the highest variance in the subspace spanned by the 37 face annotations, which is embedded in a 2×58 dimensional space. The most dominant deformation is the upward movement of the upper jaw together

²The maximal correlation between two sets of variables when general linear mappings on both are allowed.



Fig. 3. Unaligned shapes (left). Aligned shapes (right).

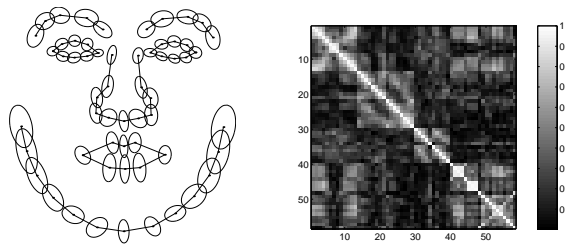


Fig. 4. Principal point directions (left). The canonical correlation of landmarks (right).

with a downward movement of nose and mouth, and vice versa. When visualising this deformation as a movie sequence, it is clear that a major part of the variation picked up in this direction in the hyperspace does not correspond to inter-subject variation in head anatomy. It is merely due to changes in the projection of the 3D landmarks into the 2D image plane, stemming from the head posture.

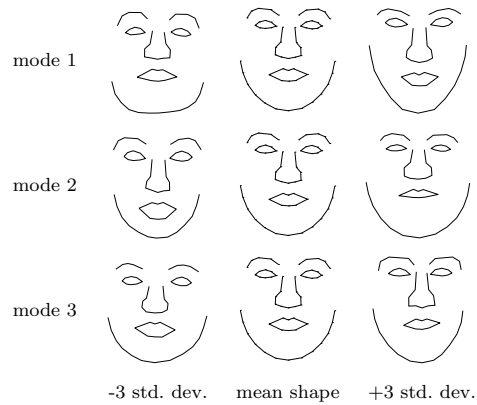


Fig. 5. Shape deformation using the three largest principal modes (row-wise, top-down).

The amount of variance explained by the ten largest eigenvalues is shown in Table II. While the three modes of Figure 5 covered 58% variation, it requires 20 modes to retain 95% of the total shape variation present in the training set.

TABLE II
TEN LARGEST EIGENVALUES OF THE SHAPE MODEL

Mode	Variance	Acc. variance
1	37.34%	37.34%
2	12.66%	50.00%
3	8.22%	58.22%
4	5.92%	64.14%
5	4.64%	68.77%
6	4.32%	73.10%
7	3.45%	76.55%
8	2.69%	79.24%
9	2.43%	81.67%
10	2.18%	83.85%

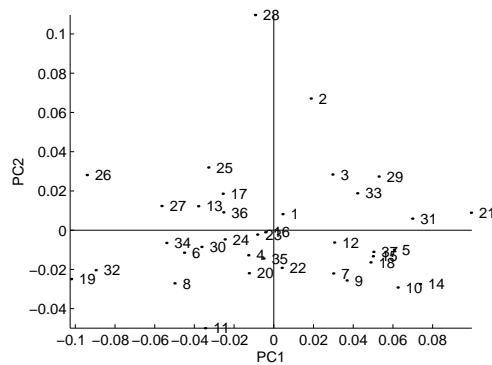


Fig. 6. PC1 ($b_{s,1}$) versus PC2 ($b_{s,2}$) in the shape PCA.

To examine if any outliers are included into the shape model, all 37 faces are projected onto the first and second shape mode in Figure 6. In this face study, observation number 28 is revealed as an outlier in principal component two. However in this case, this underlines the limited training set size rather than indicating an abnormal jaw growth.

Finally, the principal scores of the five largest shape modes are inspected for non-linearities in the scatter plot in Figure 7 to ensure a proper (i.e. specific) shape representation. Figure axes are scaled accordingly to the variance of each parameter, thus depicting Gaussian distributions as circular symmetric densities. From this figure, the basis is accepted as not being over-complete, i.e. a generative multivariate Gaussian model will cover the face space sufficiently well and not generate implausible face shapes.

A.2 Texture Model

The face texture model for all 37 faces is built from 31221 sample positions in the reference frame, which is the Procrustes mean shape sized to mean size. The sample positions are obtained by sampling

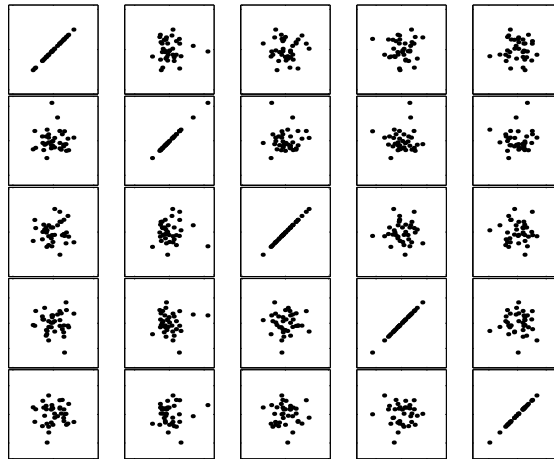


Fig. 7. Scatter plot of the five largest shape principal components (PC1 is top-left). Axes are set to $\pm 4.5\sigma$ for each parameter.

the reference frame in a grid aligned with the x and y axis and with one-pixel spacing between grid points. At each position a red, green and blue sample was obtained, resulting in a colour texture vector of 93663 samples.

Table III shows the ten largest eigenvalues of the texture PCA model. The three largest of these, accounting for 37% variation, are visualised as deformations of the mean texture in Figure 9, \pm three standard deviations. Retaining 95% of the texture variation in the training set requires 29 modes.

To inspect the training set for texture outliers all 37 faces are projected onto the first and second principal axes in Figure 8. From the first mode in Figure 9 we see that Figure 8 can be used to determine the degree of "beardedness" in faces in this case.

TABLE III
TEN LARGEST EIGENVALUES OF THE TEXTURE MODEL

Mode	Variance	Acc. variance
1	19.80%	19.80%
2	9.58%	29.38%
3	7.61%	36.99%
4	6.52%	43.51%
5	5.69%	49.20%
6	4.71%	53.91%
7	4.15%	58.06%
8	3.50%	61.56%
9	3.29%	64.84%
10	2.96%	67.81%

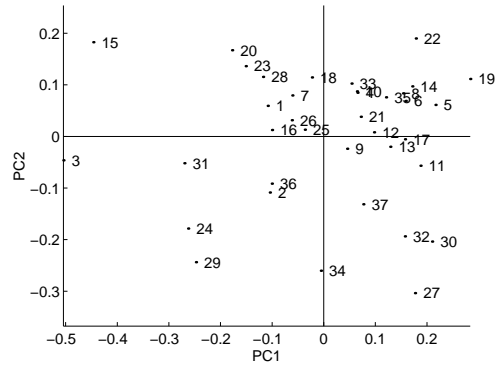


Fig. 8. PC1 ($b_{t,1}$) versus PC2 ($b_{t,2}$) in the texture PCA (right).

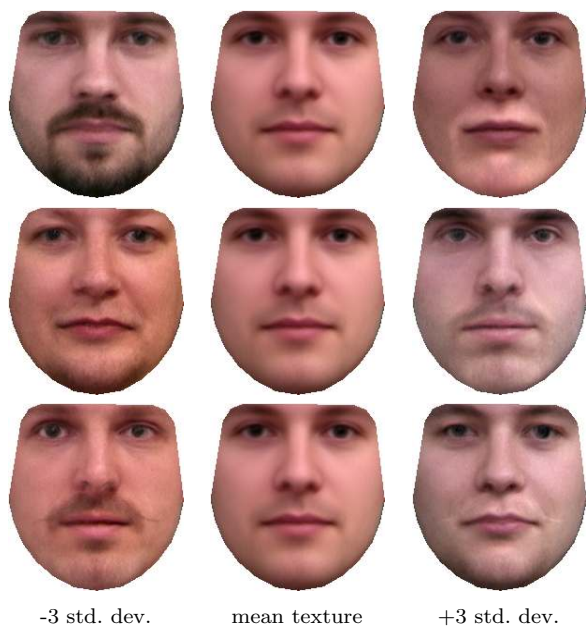


Fig. 9. Texture deformation using the three largest principal modes (row-wise, top-down).

A.3 Appearance Model

Traditional AAMs use coupled eigenspaces of shape and texture by combining the PC scores in a third PC analysis. The ten largest eigenvalues are shown in Table IV. The corresponding three largest deformation modes of shape and texture are shown in Figure 10. Refer to FAME tool *AAMLab* for real-time exploration of the modes of the combined appearance model.

A.4 Model Training

In Section IV two methods for model training were summarised. This section aims at assessing the quality of each of these methods for predicting AAM parameters. Both hypothesised a simple linear

TABLE IV
TEN LARGEST EIGENVALUES OF THE COMBINED MODEL

Mode	Variance	Acc. variance
1	22.74%	22.74%
2	12.59%	35.33%
3	7.82%	43.16%
4	5.81%	48.96%
5	5.17%	54.13%
6	4.29%	58.42%
7	4.00%	62.42%
8	3.42%	65.84%
9	3.14%	68.98%
10	2.94%	71.91%

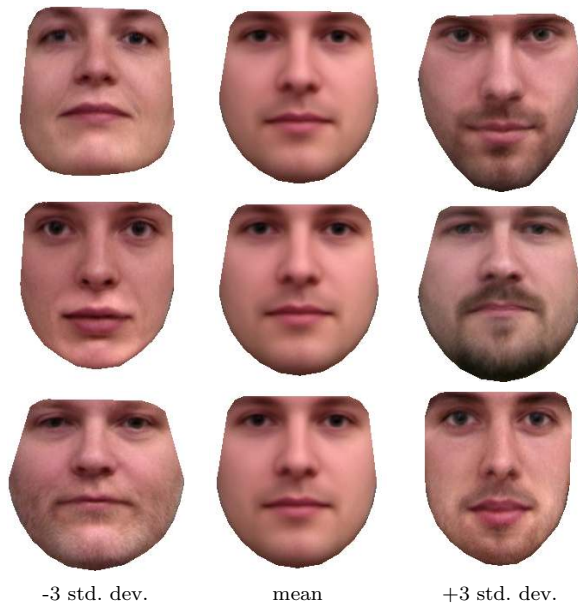
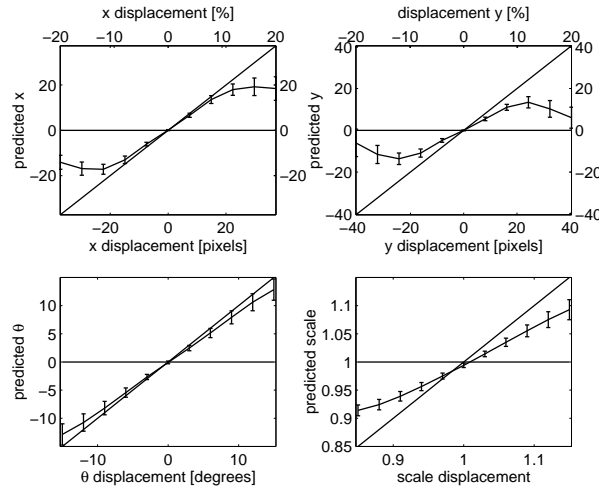


Fig. 10. Combined deformation using the three largest principal modes (row-wise, top-down).

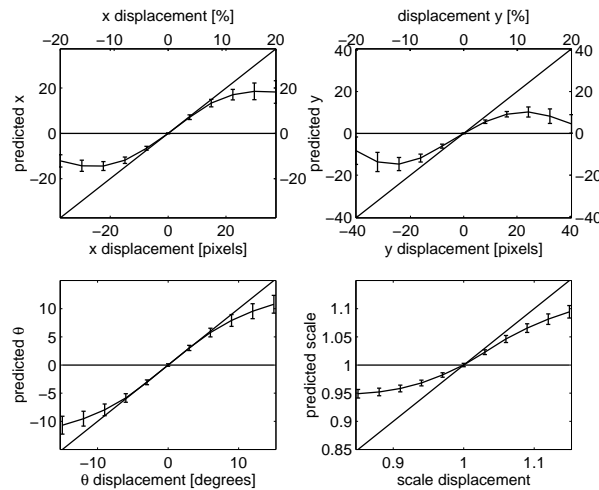
relationship between a residual vector and needed parameter updates. Desirable properties of such a prediction matrix include i) ability to predict the displacements learned, ii) ability to inter- and extrapolate the displacements learned, and most importantly iii) high prediction accuracy around zero.

To assess the pose prediction abilities, all 37 training shapes were systematically displaced in all pose parameters, one by one. Results for both learning methods are shown in Figure 11. Error bars are one std. dev. In all cases only four images was used to train the models (the first image and then every tenth). These plots provide no significant evidence for choosing one learning method over the other. Consequently,

we suggest the Jacobian due to its substantial lower computational complexity.



(a) Colour, regression



(b) Colour, Jacobian

Fig. 11. Prediction of pose parameter updates for large models.

Due to the subsampling scheme, Figure 11 is a mixture of predictions upon known, $\frac{4}{37}$, and unknown data $\frac{33}{37}$. The next section will investigate how well these training methods generalise to unseen images with a mixture of displaced parameters.

A.5 Segmentation Accuracy

Though the generative properties of AAMs enable very sophisticated image interpretation directly from the model parameters, the most common application remains to be segmentation.

To assess the segmentation accuracy the model was initialised using the mean configuration displaced $\pm 10\%$ in x and y , relative to its width and height, i.e. four experiments per training example.

Due to the limited size of the training set, cross-validation was applied in a leave-one-out analysis leading to 37 models built from 36 examples for each evaluation. Double-mean landmark errors were calculated as the mean of all landmark points and mean of all leave-one-out experiments. Pt.pt. measures the Euclidean distance between corresponding landmarks of the model and the ground truth. Pt.crv. measures the shortest distance to the curve in a neighbourhood of the corresponding landmark. Thus, $4 \times 58 \times 37 = 8584$ pt.pt. and pt.crv. distances were calculated in each evaluation.

Results for colour and grey-scale face AAMs in two resolutions are shown in Table V and VI. AAMs in Table VI were built in 1:2 decimated versions of the training images, i.e. 320×240 pixels. These models had on average 7799 and 23398 texture samples (grey-scale and colour, respectively). All shape, texture and combined PCA models were truncated to explain 95% of the variation present in the training set, resulting in 24 combined model parameters on average. From both tables we observe that the Jacobian training scheme does not differ significantly in performance. Actually it is slightly better, which is of utmost interest due to the far smaller computational and memory demands of the Jacobian training scheme.

From Tables V and VI we also observe that the addition of colour to the models only resulted in a modest improvement of the final segmentation accuracy. However, most likely due to the gained specificity from adding colour, we have experienced that it also adds significant stability to the parameter update process (see [45]). Note the relatively small penalty in segmentation accuracy when working on 1:2 decimated images.

TABLE V
SEGMENTATION RESULTS – LARGE MODELS

Model type	Learning method	Mean pt.pt.	Mean pt.crv.
Grey-scale	Regression	6.31 \pm 1.31	3.10 \pm 0.87
Grey-scale	Jacobian	6.13 \pm 1.39	2.98 \pm 0.95
Colour	Regression	6.13 \pm 1.08	3.09 \pm 0.84
Colour	Jacobian	5.93 \pm 1.15	2.86 \pm 0.86

Due to the excessive memory consumption of the regression approach, the subsampling scheme in Section IV was rather crude. The memory usage of the Jacobian estimation does not depend on the number of training shapes. Consequently, we have tested if the more robust Jacobian estimate provided by using all training examples could result in higher segmentation accuracy. The results shown in Tables VII and VIII support this by showing a subtle increase in accuracy.

TABLE VI

SEGMENTATION RESULTS – SMALL MODELS

Model type	Learning method	Mean pt.pt.	Mean pt.crv.
Grey-scale	Regression	3.31±0.83	1.64±0.52
Grey-scale	Jacobian	3.25±0.75	1.56±0.50
Colour	Regression	3.33±0.67	1.65±0.46
Colour	Jacobian	3.15±0.61	1.50±0.45

TABLE VII

SEGMENTATION RESULTS – LARGE MODELS, NO SUBSAMPLING

Model type	Learning method	Mean pt.pt.	Mean pt.crv.
Grey-scale	Jacobian	6.03±1.37	2.91±0.96
Colour	Jacobian	5.92±1.11	2.88±0.83

TABLE VIII

SEGMENTATION RESULTS – SMALL MODELS, NO SUBSAMPLING

Model type	Learning method	Mean pt.pt.	Mean pt.crv.
Grey-scale	Jacobian	3.14±0.69	1.51±0.46
Colour	Jacobian	3.08±0.59	1.48±0.41

A.6 Model Truncation

Using parallel analysis (PA) as presented in Section V the shape and texture models can be truncated according to the scree plots shown in Figure 12. In this experiment based on all 37 examples, the first 3 shape modes and the first 10 texture modes are selected. In contrast a 95% variance threshold would select 20 and 29 modes, respectively. As expected, we observe in the very dense and highly correlated texture models that the eigenvalues stemming from parallel analysis are very different from the data eigenvalues. And vice versa for the sparser and less correlated shape models. This is not a characteristic behaviour of texture and shape models. But due to AAMs ability to recover dense correspondences with a sparse set of landmarks, and the (often occurring) manual labour involved in placing landmarks, shape models are typically much more sparse than texture models.

Having a computational perspective, the simpler model is always preferable. However w.r.t. segmentation accuracy, it could prove to be too crude, i.e. resulting in under-fitting. On the other hand the variance threshold could be too liberal, resulting in over-fitting. We have tested this by means of a leave-one-out analysis. The combined model was truncated at 95% variance as in the previous experiments.

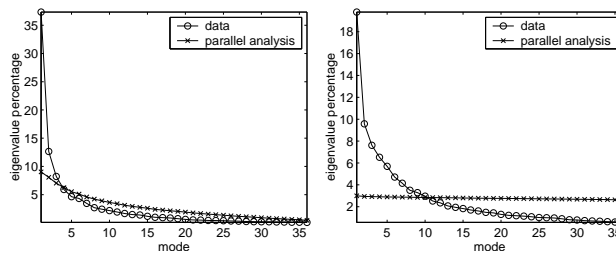


Fig. 12. Scree plots for raw (o) and randomised data (x). Shape (left), texture (right).

The results given in Tables IX and X show – in comparison with Tables VII and VIII – a modest increase in performance w.r.t. pt.pt. distance. This encouraging result suggests that parallel analysis provides both faster *and* more general AAMs in a non-parametric fashion. Consequently, simple variance thresholding seems to over-fit the training data slightly by including too many modes of variation.

TABLE IX

PA SEGMENTATION RESULTS – LARGE MODELS, NO SUBSAMPLING

Model type	Learning method	Mean pt.pt.	Mean pt.crv.
Grey-scale	Jacobian	5.70 ± 1.18	3.02 ± 0.76
Colour	Jacobian	5.54 ± 1.17	2.93 ± 0.78

TABLE X

PA SEGMENTATION RESULTS – SMALL MODELS, NO SUBSAMPLING

Model type	Learning method	Mean pt.pt.	Mean pt.crv.
Grey-scale	Jacobian	2.87 ± 0.62	1.50 ± 0.39
Colour	Jacobian	2.79 ± 0.60	1.47 ± 0.40

B. Cardiac Magnetic Resonance Images

For the second case study, short-axis, end-diastolic cardiac MRIs were selected from 14 subjects. MRIs were acquired using a whole-body MR unit (Siemens Impact) operating at 1.0 Tesla. The chosen MRI slice position represented low morphologic complexity, judged by the presence of papillary muscles. Images were acquired using an ECG-triggered breath-hold fast low angle shot (FLASH) cinematographic pulse sequence. Slice thickness=10 mm; field of view=263×350 mm; matrix 256×256. The endocardial and epicardial contours of the left ventricle were annotated manually by placing 33 landmarks along each contour, see Figure 13. To fix rotation around the left ventricular (LV) long-axis, the right ventricle (RV) was annotated using 12 landmarks. Prior to further processing, subject 11 was removed due to a scanner

calibration error.

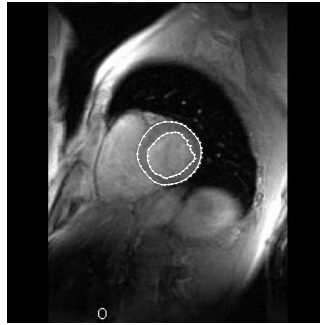


Fig. 13. Example annotation of the left ventricle using 66 landmarks.

B.1 Shape, Texture and Combined Model

Figure 14 shows all cardiac shapes unaligned (left) and Procrustes aligned (right). Combining the cardiac shape and texture model yields the first three modes of combined variation shown in Figure 15. These three modes account for 22%, 20%, 14% of the total variance, respectively.

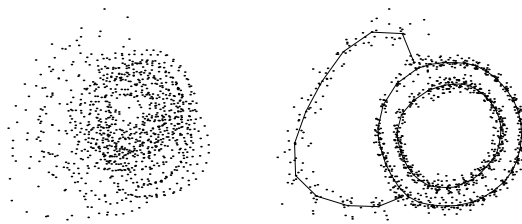


Fig. 14. Unaligned shapes (left). Aligned shapes (right).

B.2 Segmentation Accuracy

To assess segmentation accuracy an evaluation methodology similar to that of the face case was adopted. The models stemming from the leave-one-out experiments contained 4192 texture samples on average. Variance-based truncation at a 95% level, yielded 9 combined parameters, whereas parallel analysis settled on 5 parameters on average. No subsampling was used during the training phase. The corresponding results are summarised in Table XI and Table XII using both learning methods. Consistent with the face case we observe that i) parallel analysis provided slightly better accuracy using fewer model modes and ii) no significant difference between the two learning methods existed. The average segmentation time was 29 ms.

FAME also provides a generic method for unsupervised model initialisation. Basically, this is a semi-brute force model-based search in selected model and pose parameters, which exploits the convergence

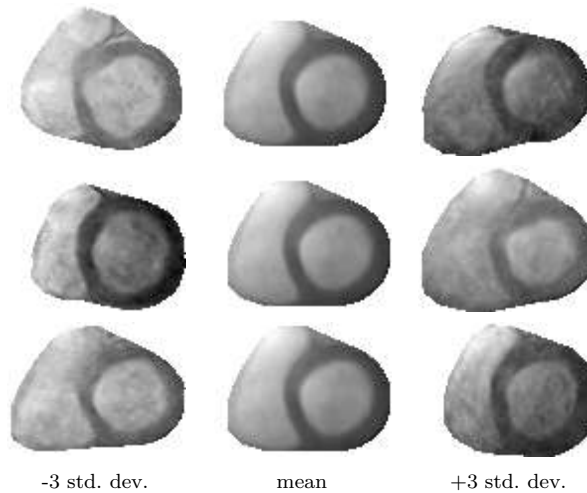


Fig. 15. Combined deformation using the three largest principal modes (row-wise, top-down).

TABLE XI

SEGMENTATION RESULTS – VARIANCE TRUNCATION

Learning method	Mean pt.pt.	Mean pt.crv.
Regression	3.27 ± 1.06	1.38 ± 0.34
Jacobian	3.20 ± 0.91	1.28 ± 0.25

TABLE XII

SEGMENTATION RESULTS – PARALLEL ANALYSIS

Learning method	Mean pt.pt.	Mean pt.crv.
Regression	3.02 ± 0.83	1.38 ± 0.31
Jacobian	3.11 ± 0.91	1.36 ± 0.26

radius of each parameter. To add robustness this is implemented in a candidate scheme, where several initialisation candidates compete to become the initial configuration. For details see [46].

This method has been employed in leave-one-out experiments similar to the above. The results given in Table XIII show good correspondence with Table XI, suggesting that unsupervised segmentation is plausible without employing a dedicated initialisation method.

Finally, the alleged robustness to image noise was tested by adding Gaussian noise at different levels, $\sigma^2 \in [0; 40^2]$, to the training set. Image intensities were in the range $[0; 255]$. For each level a leave-one-out study similar to the above was conducted. To improve the robustness we have used the multi-scale feature that FAME provides. Here optimisation is started at the smallest level of a pyramidal set of AAMs and

TABLE XIII
 SEGMENTATION USING AUTOMATIC MODEL INITIALISATION
 (USING VARIANCE TRUNCATION)

Learning method	Mean pt.pt.	Mean pt.crv.
Regression	3.20 ± 1.18	1.31 ± 0.30
Jacobian	3.21 ± 1.02	1.28 ± 0.24

then propagated down towards the original resolution. The results given in Figure 16 show a modest 15% increase in pt.crv. accuracy when comparing the original images to the most noise contaminated. Further, we observe that the accuracy degrades gracefully with increasing noise amplitude. Figure 17 shows an average segmentation result with $\sigma = 40$.

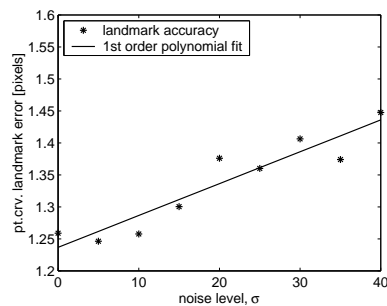


Fig. 16. AAM segmentation accuracy in the presence of noise.

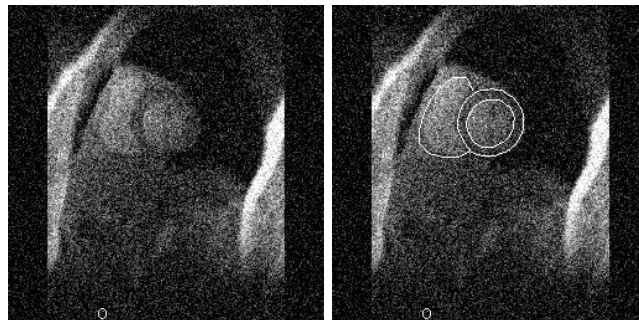


Fig. 17. Cardiac MRI with added noise, $\sigma = 40$ (left). Corresponding segmentation result.

This concludes the study of cross-sectional cardiac MRI. Here we have treated segmentation of both LV and RV. We note that typically only the segmentation of LV is of interest in, e.g. an ejection fraction study. In this case a hierarchical approach [47] should be used to increase segmentation accuracy. Here, an LV model should be de-coupled from the converged RV-LV model and iterated further, thus relaxing model constraints. See also [33] on this matter.

VIII. DISCUSSION AND CONCLUSION

In this paper we have described an appearance-based software environment for interpretation of images. This is the Flexible Appearance Modelling Environment (FAME), which is based on the well-known Active Appearance Model (AAM) framework. We have summarised the wide range of medical applications that frameworks similar to ours have been applied to. In particular, we have treated two crucial parts in the building process of an AAM. Namely, how to learn the update step used during model fitting, and how to choose the number of model parameters. These issues along with several others have been evaluated in two different case studies upon short-axis cardiac magnetic resonance images and face images. The proposed unsupervised method for choosing the number of model modes proved preferable in all cases. Further, it has been verified that the Jacobian learning scheme is preferable to the regression approach taken in the original AAM formulation. In all studies the AAM variation implemented in FAME has proven accurate, fast and very robust to image noise. An example of how a strong prior redeems an otherwise ill-posed segmentation problem was given in the Cardiac MRI study. This showed a very modest penalty in segmentation accuracy even for very noisy data when using a multi-scale approach.

The performance provided by the framework renders it feasible to interactive analysis of medical images. Further, it allows the usage of general-purpose optimisers to "fine-polish" the converged search results in a reasonable amount of time [25], [33]. For completely unsupervised image analysis, FAME also provides means of automatic model segmentation. Due to the modular structure of FAME, it is straightforward to extend it to cope with other medical data than 2D images, e.g. time-series of 2D images, coupled images (e.g. from different imaging modalities) etc.

The FAME package including source code and all data material used in this paper has been placed in the public domain. Thus we encourage researchers to apply, extend and embed FAME in future studies both medical and non-medical. Further, we advocate for a general trend between researchers to benchmark their methods against reference data sets and reference implementations, thus providing a much more clear view on the progress in the field of medical image analysis.

ACKNOWLEDGMENTS

The following are gratefully acknowledged for their help in this work. Cardiac MRIs were provided by M.D., Ph.D., J. C. Nilsson and M.D., Ph.D., B. A. Grønning, Danish Research Centre for Magnetic Resonance, DRCMR, Hvidovre University Hospital. M. M. Nordstrøm, M. Larsen and J. Sierakowski collected and annotated the face images. We would further like to thank the anonymous reviewers for their helpful comments on an earlier version of this manuscript.

APPENDIX

I. NOTES ON THE FAME IMPLEMENTATION

Texture Synthesis

As shown above, texture synthesis is carried out by

$$\mathbf{t} = \bar{\mathbf{t}} + \Phi_t \Phi_{c,t} \mathbf{c}. \quad (14)$$

For clarity reasons this expression is often written in the AAM literature as

$$\mathbf{t} = \bar{\mathbf{t}} + \Phi \mathbf{c}. \quad (15)$$

Let,

$$k_t = \text{rank}(\Phi_t) \quad , \quad k_c = \text{rank}(\Phi_{c,t}) \quad (16)$$

Though it seems tempting to pre-calculate the constant matrix product in (14), this should only be done when $M(k_c - k_t) < k_t k_c$. Otherwise, synthesis should be carried out as

$$\mathbf{t} = \bar{\mathbf{t}} + \Phi_t (\Phi_{c,t} \mathbf{c}). \quad (17)$$

To illustrate this, consider the AAM described in [48] having 66 shape parameters and 10 texture parameters and 5040 texture samples. If we assume that these are only moderately correlated, 74 combined parameters would seem plausible. The extra cost of using (15) compared to (14) is $M(k_c - k_t) - k_t k_c$. Consequently, this amounts to 321820 extra multiplications for each texture synthesis.

Texture Normalisation

Contrary to e.g. [37], [36], [7] FAME does not include a model for scaling and offset of textures. During analysis texture vectors are standardised to zero mean and fixed variance. During synthesis, model textures are fitted to image textures in a least squares sense.

Pose parameterisation

To estimate the Jacobian matrix, pose parameters need to be independent. To obtain equilibrium at zero we use the following representation of pose

$$\boldsymbol{\rho} = [(s - 1) \theta \ t_x \ t_y]^T, \quad (18)$$

where s denotes scale, θ orientation and t_x, t_y in-plane translation.

II. FAME MODEL OPTIONS

This section enumerates the most important model design options offered by FAME.

- *Model reduction* Determines the image reduction prior to model building, i.e. restricts the number of texture samples in the model.
- *Convex hull* Determines whether i) all texture samples inside the convex hull of the mean shape should be modelled or ii) only the texture samples inside closed contours.
- *Tangent space projection* Determines if a linearisation of the shape manifold around the mean pole should be carried out.
- *Learning method* Selects either Principal Component Regression or Jacobian Estimation to build parameter update matrices.
- *Training set subsampling* Option that chooses every n -th example in the training set to build parameter update matrices upon.
- *Model truncation* Controls the number of parameters in the shape, texture and combined models. Either i) by a variance threshold, or ii) using parallel analysis to estimate the number of modes to retain.
- *Warping method* Selects i) software warping, ii) hardware warping, or iii) the benchmarking mode, where the model building process is replaced by a performance test of the two warping methods.

III. HARDWARE-ASSISTED AAM WARPING

Contemporary graphics hardware is highly optimised for performing piece-wise affine warps. Below we will demonstrate how to exploit this in an AAM framework.

In AAMs two types of warps are carried out. One during the analysis phase (i.e. model building and model optimisation) and a one during synthesis. In the analysis phase many-to-one warps are carried out. Different configurations of the shape model are all warped to same shape-free reference frame, which is typically the mean shape sized to mean size. During synthesis shape-free textures are warped to specific shape configurations.

We approach the graphics hardware through the industry standard for graphics programming: OpenGL. Here a triangular mesh can be defined and images can be projected onto this surface by means of texture mapping. In addition, recent extensions such as hardware-accelerated off-screen buffers, non-dyadic textures etc. can be utilised. For details on the implementation and a discussion of its bottlenecks, refer to [45].

IV. DETAILS ON MACHINERY AND COMPUTATION TIME

To give an impression of the applicability of these models timings were acquired for the face model building phase and the optimisation phase. These are given as reference numbers. For maximal speed a multi-resolution model should be used. The build timings were calculated for both large and small models in colour and grey-scale built on all 37 training examples using the subsampling scheme mentioned in Section IV. All timings in Table XV were obtained on similar models but using Jacobian and leave-one-out on the training examples.

All results shown above including the performance measures in Table XIV and XV were obtained on an Athlon 1200 MHz equipped with 768 MB RAM using software warping. Except for the last row in Table XIV that used a GeForce 2 MX for hardware warping. From Tables XIV and XV we can observe that warping is not the bottleneck in the current implementation.

In Table XVI we used a Pentium 4 mobile 1700 Mhz equipped with an NVidia GeForce 4 Go graphics board. We have used high performance timers and several hundred repetitions in order to measure accurately in the millisecond range. From this table we can observe that the specific graphics board (and its driver) has a great impact on the performance. Further, since the hardware AAM warping is very heavy on bus i/o the infrastructure between GPU and CPU is of utmost importance.

TABLE XIV

MODEL BUILDING AND WARPING (SMALL/LARGE MODELS)

	Gray-scale	Colour
Model build, regression	0:31 / 1:11 mins	1:03 / 2:51 mins
Model build, Jacobian	0:09 / 0:39 mins	0:27 / 1:54 mins
Software analysis warp	1 / 9 ms	3 / 12 ms
Hardware analysis warp	2 / 5 ms	2 / 7 ms

TABLE XV

OPTIMISATION TIMINGS (SMALL/LARGE MODELS)

	Gray-scale	Colour
Mean number of iterations	11.4 / 11.3	8.7 / 8.2
Mean optimisation time	129 / 581 ms	281 / 1253 ms
1 optimisation iteration	11 / 51 ms	32 / 153 ms

TABLE XVI

WARP TIMINGS – GEFORCE 4 GO (LARGE MODEL)

	Gray-scale	Colour
Software analysis warp	5.4 ms	7.4 ms
Hardware analysis warp	1.5 ms	2.4 ms

REFERENCES

- [1] G. J. Edwards, C. J. Taylor, and T. F. Cootes, "Interpreting face images using active appearance models," in *Proc. 3rd IEEE Int. Conf. on Automatic Face and Gesture Recognition*. 1998, pp. 300–5, IEEE Comput. Soc.
- [2] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *Proc. European Conf. on Computer Vision*. 1998, vol. 2, pp. 484–498, Springer.
- [3] T. F. Cootes, C. Beeston, G. J. Edwards, and C. J. Taylor, "A unified framework for atlas matching using active appearance models," in *Information Processing in Medical Imaging. IPMI'99*. 1999, pp. 322–33, Springer-Verlag.
- [4] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Comparing active shape models with active appearance models," in *Proc. British Machine Vision Conf.*, 1999, pp. 173–182.
- [5] C. B. H. Wolstenholme and C. J. Taylor, "Wavelet compression of active appearance models," in *Medical Image Computing and Computer-Assisted Intervention, MICCAI*, 1999, pp. 544–554.

- [6] T. F. Cootes and C. J. Taylor, “Combining elastic and statistical models of appearance variation,” in *Proc. European Conf. on Computer Vision*, 2000, vol. 1, pp. 149–163.
- [7] T. F. Cootes and C. J. Taylor, “Statistical models of appearance for medical image analysis and computer vision,” in *Proc. SPIE Medical Imaging 2001*. 2001, vol. 1, pp. 236–248, SPIE.
- [8] T. F. Cootes and C. J. Taylor, *Statistical Models of Appearance for Computer Vision*, Tech. Report, University of Manchester, 2000.
- [9] S. Mitchell, B. Lelieveldt, R. Geest, J. Schaap, J. Reiber, and M. Sonka, “Segmentation of cardiac MR images: An active appearance model approach,” in *Medical Imaging 2000: Image Processing, San Diego CA, SPIE*. 2000, vol. 1, SPIE.
- [10] S. C. Mitchell, B. P. F. Lelieveldt, R. J. van der Geest, H. G. Bosch, J. H. C. Reiver, and M. Sonka, “Multistage hybrid active appearance model matching: segmentation of left and right ventricles in cardiac MR images,” *Medical Imaging, IEEE Transactions on*, vol. 20, no. 5, pp. 415–423, 2001.
- [11] S. Mitchell, B. Lelieveldt, R. Geest, H. Bosch, J. Reiber, and M. Sonka, “Time continuous segmentation of cardiac MR image sequences using active appearance motion models,” in *Medical Imaging 2001: Image Processing, San Diego CA, SPIE*. 2001, vol. 1, pp. 249–256, SPIE.
- [12] B. P. F. Lelieveldt, S. C. Mitchell, R. J. van der Geest, H. G. Bosch, M. Sonka, and J. H. C. Reiber, “Time continuous segmentation of cardiac MR images using active appearance motion models,” in *Computer Assisted Radiology and Surgery, CARS 2001, International Congress Series*. 2001, vol. 1230, pp. 961–966, Elsevier Science.
- [13] M. Sonka, B. P. F. Lelieveldt, S. C. Mitchell, J. G. Bosch, R. J. van der Geest, and J. H. C. Reiber, “Active appearance motion model segmentation,” *Proceedings Second International Workshop on Digital and Computational Video*, pp. 64–8, 2001.
- [14] H. G. Bosch, S. C. Mitchell, B. P. F. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, and J. H. C. Reiber, “Active appearance-motion models for endocardial contour detection in time sequences of echocardiograms,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 4322, no. 1, pp. 257–268, 2001.
- [15] J. G. Bosch, S. C. Mitchell, B. P. F. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, and J. H. C. Reiber, “Fully automated endocardial contour detection in time sequences of echocardiograms by active appearance motion models,” *Computers in Cardiology*, pp. 93–96, 2001.
- [16] S. C. Mitchell, B. P. Lelieveldt, J. G. Bosch, R. van der Geest, J. H. Reiber, and M. Sonka, “Segmentation of cardiac MR volume data using 3D active appearance models,” *Medical Imaging 2002: Image Processing, San Diego CA, SPIE*, pp. 433–443, 2002.
- [17] S. C. Mitchell, J. G. Bosch, B. P. F. Lelieveldt, R. J. van der Geest, J. H.C. Reiber, and M. Sonka, “3-D active appearance models: Segmentation of cardiac MR and ultrasound images,” *Medical Imaging, IEEE Transactions on*, vol. 21, no. 9, pp. 1167–1178, 2002.
- [18] B. Lelieveldt, R. van der Geest, S. C. Mitchell, J. Bosch, M. Sonka, and J. Reiber, “3D active appearance models: Fully automatic detection of endo- and epicardial contours in short-axis cardiac MR data,” in *Proc. International Society of Magnetic Resonance In Medicine, ISMRM*. 2002, vol. 2, p. 1668, ISMRM.
- [19] R. Beichel, S. Mitchell, E. Sorantin, F. Leberl, A. Goshtasby, and M. Sonka, “Shape- and appearance-based segmentation of volumetric medical images,” *IEEE International Conference on Image Processing*, vol. 2, pp. 589–592, 2001.
- [20] R. Beichel, G. Gotschuli, E. Sorantin, F. W. Leberl, and M. Sonka, “Diaphragm dome surface segmentation in ct data sets: a 3D active appearance model approach,” *Medical Imaging 2002: Image Processing, San Diego CA, SPIE*, pp. 475–484, 2002.
- [21] J. G. Bosch, S. C. Mitchell, B. P. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, and J. H. Reiber, “Fully automated endocardial contour detection in time sequences of echocardiograms by three-dimensional active appearance models,” *Medical Imaging 2002: Image Processing, San Diego CA, SPIE*, pp. 452–462, 2002.
- [22] S. Duchesne, J. C. Pruessner, and D. L. Collins, “Appearance-based modelling and segmentation of the hippocampus from MR images,” *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, vol. 3, pp. 2677–2680, 2001.

- [23] M. J. Jones and T. Poggio, “Multidimensional morphable models: a framework for representing and matching object classes,” *International Journal of Computer Vision*, vol. 29, no. 2, pp. 107–31, 1998.
- [24] M. B. Stegmann, “Active appearance models: Theory, extensions and cases,” M.S. thesis, Informatics and Mathematical Modelling, Technical University of Denmark, Lyngby, 2000, <http://www.imm.dtu.dk/~aam/>.
- [25] M. B. Stegmann, R. Fisker, and B. K. Ersbøll, “Extending and applying active appearance models for automated, high precision segmentation in different image modalities,” in *Proc. 12th Scandinavian Conference on Image Analysis - SCIA 2001*, 2001, vol. 1, pp. 90–97.
- [26] M. B. Stegmann, J. C. Nilsson, and B. A. Grønning, “Automated segmentation of cardiac magnetic resonance images,” in *Proc. International Society of Magnetic Resonance In Medicine - ISMRM 2001*, 2001, vol. 2, p. 827, ISMRM.
- [27] M. B. Stegmann and S. Forchhammer, “On exploiting wavelet bases in statistical region-based segmentation,” in *Proc. 11th Danish Conference on Pattern Recognition and Image Analysis*, Copenhagen, Denmark, aug 2002, vol. 1, pp. 75–82, DIKU.
- [28] H. H. Thodberg, “Hands-on experience with active appearance models,” in *Medical Imaging 2002: Image Processing, San Diego CA, SPIE*, 2002, pp. 495–506, SPIE.
- [29] M. B. Stegmann and H. B. W. Larsson, “Fast registration of cardiac perfusion MRI,” in *Proc. International Society of Magnetic Resonance In Medicine – ISMRM 2003, Toronto, Ontario, Canada*, Berkeley, CA, USA, 2003, p. 702, ISMRM.
- [30] M. B. Stegmann and H. B. Larsson, “Motion-compensation of cardiac perfusion MRI using a statistical texture ensemble,” in *Functional Imaging and Modeling of the Heart, FIMH 2003*, Lyon, France, 2003, vol. 2674 of *LNCS*, pp. 151–161, Springer.
- [31] K. B. Hilger, M. B. Stegmann, and R. Larsen, “A noise robust statistical model for image representation,” in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2002, 5th Int. Conference, Tokyo, Japan*, sep 2002, vol. 2488 of *LNCS*, pp. 444–451.
- [32] M. B. Stegmann and R. H. Davies, “Automated analysis of corpora callosa,” Tech. Rep. IMM-REP-2003-02, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, <http://www.imm.dtu.dk/~mbs/>, mar 2003.
- [33] M. B. Stegmann, R. H. Davies, and C. Rybjerg, “Corpus callosum analysis using MDL-based sequential models of shape and appearance,” in *(preparation)*, 2003.
- [34] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models – their training and application,” *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.
- [35] J. C. Gower, “Generalized Procrustes analysis,” *Psychometrika*, vol. 40, pp. 33–50, 1975.
- [36] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” *IEEE Trans. on Pattern Recognition and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.
- [37] T. F. Cootes and C. J. Taylor, “Constrained active appearance models,” *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, pp. 748–754 vol.1, 2001.
- [38] T. F. Cootes and P. Kittipanya-ngam, “Comparing variations on the active appearance model algorithm,” in *Proceedings of the British Machine Vision Conference, BMVC, 2002*, vol. 2, pp. 837–846.
- [39] B. Efron, “Bootstrap methods: another look at the jackknife,” *Annals of Statistics*, , no. 7, pp. 1–26, 1979.
- [40] J. L. Horn, “A rationale and test for the number of factors in factor analysis,” *Psychometrika*, vol. 30, pp. 179–186, 1965.
- [41] Microsoft, “Vision SDK,” 2000, Web page: <http://research.microsoft.com/projects/VisSDK/>.
- [42] LAPACK, “Lapack – linear algebra package,” Web page: <http://www.netlib.org/lapack/>.
- [43] T. J. Hutton, B. R. Buxton, and P. Hammond, “Dense surface point distribution models of the human face,” *Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA 2001)*, pp. 153–60, 2001.
- [44] A. F. Frangi, W. J. Niessen, and M. A. Viergever, “Three-dimensional modeling for functional analysis of cardiac images: A review,” *IEEE Transactions on Medical Imaging*, vol. 20, no. 1, pp. 2–25, 2001.
- [45] M. B. Stegmann, “Analysis and segmentation of face images using point annotations and linear subspace techniques,” Tech. Rep. IMM-REP-2002-22, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, <http://www.imm.dtu.dk/~mbs/>, aug 2002.

- [46] M. B. Stegmann, “Object tracking using active appearance models,” in *Proc. 10th Danish Conference on Pattern Recognition and Image Analysis, Copenhagen, Denmark*. 2001, vol. 1, pp. 54–60, DIKU.
- [47] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3D faces,” *Computer Graphics Proceedings. SIG-GRAPH 99*, pp. 187–94, 1999.
- [48] J. Ahlberg, *Model-based Coding – Extraction, Coding and Evaluation of Face Model Parameters*, Ph.D. thesis, Department of Electrical Engineering, Linköping University, Sweden, 2002.