# EXTENDING AND APPLYING ACTIVE APPEARANCE MODELS FOR AUTOMATED, HIGH PRECISION SEGMENTATION IN DIFFERENT IMAGE MODALITIES

*Mikkel B. Stegmann, Rune Fisker, Bjarne K. Ersbøll*

Informatics and Mathematical Modelling, Technical University of Denmark
DTU Building 321, DK-2800 Lyngby, Denmark

## ABSTRACT

In this paper, we present a set of extensions to the deformable template model: Active Appearance Model (AAM) proposed by Cootes et al. AAMs distinguish themselves by learning a priori knowledge through observation of shape and texture variation in a training set. This is used to obtain a compact object class description, which can be employed to rapidly search images for new object instances. The proposed extensions concern enhanced shape representation, handling of homogeneous and heterogeneous textures, refinement optimization using Simulated Annealing and robust statistics. Finally, an initialization scheme is designed thus making the usage of AAMs fully automated. Using these extensions it is demonstrated that AAMs can segment bone structures in radiographs, pork chops in perspective images and the left ventricle in cardiovascular magnetic resonance images in a robust, fast and accurate manner. Subpixel landmark accuracy was obtained in two of the three cases.

**Keywords:** Deformable Template Models, Snakes, Robust Statistics, Initialization, Metacarpal Radiographs, Cardiovascular Magnetic Resonance Imaging, Segmentation.

## 1. INTRODUCTION

In recent years, the model-based approach towards image interpretation named deformable template models has proven very successful. This is especially true in the case of images containing known objects with large variability.

Among the earliest and most well known deformable template models is the Active Contour Model – known as *Snakes* proposed by Kass et al. [20]. Snakes represent objects as a set of outline landmarks upon which a correlation structure is forced to constrain local shape changes. In order to improve specificity, many attempts at hand crafting a priori knowledge into a deformable template model have been carried out. These include parameterization of a human eye using ellipsis and arcs by Yuille et al. [28].

In a more general approach, while preserving specificity Cootes et al. [7] proposed the Active Shape Models (ASM) where shape variability is learned through experimental observation. In practice, this is accomplished by a training set of annotated examples followed by a Procrustes analysis combined with a principal component analysis.

A direct extension of the ASM approach has lead to the Active Appearance Models (AAMs) [4]. Besides shape information, the textural information, i.e. the pixel intensities across the object, is included into the model. AAMs have been further developed in [5, 6, 9, 23]. Quite similar to AAMs and developed in parallel herewith, Sclaroff & Isidoro suggested the Active Blob approach [24]. Active Blobs is a real-time tracking technique, which captures shape and textual information from a prototype image using a finite element model, to model shape variation. Compared to AAMs, Active Blobs deform a static texture, whereas AAMs deforms both texture and shape during the optimization. Early modeling of texture includes the eigenfaces by Turk & Pentland [27], where face recognition was accomplished using a PCA-based texture model similar to the one integrated into AAMs.

Other general deformable template models include the ones proposed by Grenander [15] and Jain [19]. For further information on deformable template models, refer to the surveys given in [10, 18, 22].

## 2. ACTIVE APPEARANCE MODELS

Below is presented the outline of the Active Appearance Model approach. AAMs distinguish themselves from many other segmentation methods in the sense that segmentation can be carried out using the approach as a black box. The user only needs to provide a training set of annotated shapes. For further details refer to [4, 6, 26].

### 2.1. Shape & Landmarks

AAMs handle shapes as a finite set of landmarks. Here the term shape is defined as *"All the geometrical information that remains when location, scale and rotational effects are filtered out from an object."* [8] and the concept of a landmark as *"A point of correspondence on each object that matches between and within populations."* [8].

A mathematical representation of a shape with $n$-points in $k$ dimensions could be a concatenation of each dimension

Corresponding author: M. B. Stegmann, aam@imm.dtu.dk
Project web-site: http://www.imm.dtu.dk/∼aam/

in a $kn$-vector. The vector representation used for planar shapes is then:

$$\mathbf{x} = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)^{\mathrm{T}} \quad (1)$$

Notice that the above representation does not contain any explicit information about point connectivity. In the presented framework, point connectivity is added as auxiliary data.

### 2.2. Shape Formulation

When dealing with redundancy in multivariate data – such as shapes – AAMs utilize the linear orthogonal transformation; *principal component analysis* (PCA). In our application for describing shape variation by PCA – a shape of $n$ points is considered one observation, $\mathbf{x}_i$, in a $2n$ dimensional space.

In practice the PCA is performed as an eigenanalysis of the covariance matrix of the shapes aligned w.r.t. position, scale and rotation, i.e. after a Procrustes analysis. As shape metric in the alignment procedure the Procrustes distance [14] is used. Other shape metrics such as the Hausdorff distance [17] could also be considered.

Consequently it is assumed that the set of $N$ shapes constitutes some ellipsoid structure of which the centroid – the mean shape – can be estimated as: $\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$. The ML estimate of the covariance matrix can thus be given as, $\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^{\mathrm{T}}$ The principal axes of the $2n^{th}$ dimensional shape ellipsoid are then given as the eigenvectors, $\mathbf{\Phi}_s$, of the covariance matrix, $\mathbf{\Sigma}$ (where $\mathbf{\Lambda}_s$ is a diagonal matrix of eigenvalues):

$$\mathbf{\Sigma}\mathbf{\Phi}_s = \mathbf{\Phi}_s\mathbf{\Lambda}_s \quad (2)$$

A new shape instance can then be generated by deforming the mean shape by a linear combination of eigenvectors, weighted by $\mathbf{b}_s$:

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{\Phi}_s\mathbf{b}_s \quad (3)$$

Essentially, the point or *nodal representation* of shape has now been transformed into a *modal representation* where modes are ordered according to the percentage of variation that they explain. To regularize and improve performance modes are included until the cumulated variation is above a certain threshold (e.g. 95%).

### 2.3. Texture Formulation

Contrary to the prevalent understanding of the term *texture* in the computer vision community, this concept will be used somewhat differently below. Here we define texture as "The pixel intensities across the object in question (if necessary after a suitable normalization)." A vector is chosen, as the mathematical representation of texture, where $m$ denotes the number of pixel samples over the object surface:

$$\mathbf{g} = (g_1, g_2, \ldots, g_m)^{\mathrm{T}} \quad (4)$$

In the shape case, the data acquisition is straightforward because the landmarks in the shape vector constitute the data itself. In the texture-case one needs a consistent method for collecting the texture information between the landmarks, i.e. an image sampling function needs to be established. This can be done in several ways. Here, a piecewise affine warp based on the Delaunay triangulation of the mean shape is used. Alternatively thin-plate splines [2] could substitute the piece-wise affine warp to obtain a smooth warp. For details on the Delaunay triangulation and image warping refer to [12, 25].

Following the warp from an actual shape to the mean shape, a normalization of the $\mathbf{g}$-vector set is performed to avoid the influence from global linear changes in pixel intensities. Hereafter, the analysis is identical to that of the shapes. Hence, a compact PCA representation is derived to deform the texture in a manner similar to what is observed in the training set:

$$\mathbf{g} = \overline{\mathbf{g}} + \mathbf{\Phi}_g\mathbf{b}_g \quad (5)$$

Where $\overline{\mathbf{g}}$ is the mean texture; $\mathbf{\Phi}_g$ represents the eigenvectors of the covariance matrix and finally $\mathbf{b}_g$ are the modal texture deformation parameters.

For all practical purposes there will always be far more dimensions in the texture vectors than observations (annotated examples) thus leading to rank deficiency in the covariance matrix. Hence, to efficiently compute the eigenvectors of the covariance matrix one must reduce the problem through use of the Eckart-Young theorem.

### 2.4. Combined Model Formulation

To remove correlation between shape and texture model parameters – and to make the model representation even more compact – a 3rd PCA is performed on the shape and texture PCA scores of the training set, $\mathbf{b}$ to obtain the combined model parameters, $\mathbf{c}$:

$$\mathbf{b} = \mathbf{Q}\mathbf{c} \quad (6)$$

The PCA scores are easily obtained due to the linear nature of the model:

$$\mathbf{b} = \begin{pmatrix} \mathbf{W}_s\mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} = \begin{pmatrix} \mathbf{W}_s\mathbf{\Phi}_s^{\mathrm{T}}(\mathbf{x} - \overline{\mathbf{x}}) \\ \mathbf{\Phi}_g^{\mathrm{T}}(\mathbf{g} - \overline{\mathbf{g}}) \end{pmatrix} \quad (7)$$

A suitable weighting between pixel distances and pixel intensities is obtained through the diagonal matrix $\mathbf{W}_s$ [6]. An alternative approach is to perform the two initial PCAs based on the correlation matrix as opposed to the covariance matrix.

Now, a complete model instance including shape, $\mathbf{x}$ and texture, $\mathbf{g}$, is generated using the $\mathbf{c}$-model parameters.

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{\Phi}_s \mathbf{W}_s^{-1} \mathbf{Q}_s \mathbf{c} \quad (8)$$

$$\mathbf{g} = \overline{\mathbf{g}} + \mathbf{\Phi}_g \mathbf{Q}_g \mathbf{c} \quad (9)$$

Regarding the compression of the model parameters, one should notice that the rank of $\mathbf{Q}$ will never exceed the number of examples in the training set.

Another feasible method to obtain the combined model is to concatenate both shape points and texture information into one observation vector from the start and then perform PCA on the correlation matrix of these observations.

### 2.5. Optimization

In AAMs the search is treated as an optimization problem in which the difference between the synthesized object delivered by the AAM and an actual image is to be minimized. By adjusting the AAM-parameters ($\mathbf{c}$ and pose) the model texture, $g_{model}$, can be deformed to fit the image, $g_{image}$, in the best possible way. The quadratic error norm is applied as optimization criterion [6]:

$$E = \sum_{i=1}^{m} (g_{model} - g_{image})^2 = \sum_{i=1}^{m} (\delta g_i)^2 = |\delta \mathbf{g}|^2 \quad (10)$$

Though the parameterization of the object class in question can be compressed markedly by the principal component analysis – by leaving out the principal axes that explain little variation – it is far from an easy task to optimize the system. This is not only computationally cumbersome but also theoretically challenging since it is not guaranteed that the search-hyperspace is convex. However, AAMs circumvent these potential problems in a rather untraditional fashion, assuming a linear relationship between parameter changes, $\delta\mathbf{c}$, and pixel differences, $\delta\mathbf{g}$.

$$\delta \mathbf{c} = \mathbf{R} \delta \mathbf{g} \quad (11)$$

Since the matrix $\mathbf{R}$ is estimated once at model building time, this is very run-time efficient by avoiding any computationally expensive and potentially unstable high-dimensional optimization. In practice $\mathbf{R}$ is estimated by a set of experiments using the training set, which are fed into a multivariate principal regression framework. In the AAM optimization, this prediction scheme is applied iteratively. Fig. 1 shows a prediction plot for one pose parameter. For details refer to [6, 26]. It should be noticed that the Active Blobs approach [24] is optimized using a method quite similar to that of AAMs named *difference decomposition* [13].

## 3. EXTENSIONS

### 3.1. Enhanced Shape Representation

Basic AAMs using the piece-wise affine warping, rely on the Delaunay triangulation of the shape points. This results in a triangular mesh covering the convex hull of the point set. For concave shapes this might not be the optimal solution. For example there could be substantial texture noise in the area outside the actual shape – but still inside the convex hull – thus leading to a less compact model.

To avoid this we suggest removing the triangles outside the shape. This is trivially accomplished by traversing the triangles; testing if the triangle centroid should be outside the shape polygon. If so, remove the triangle. To test if a point is inside the shape polygon we utilize the simple geometrical fact that, if a line from the point, $\mathbf{p}$, to infinity crosses the polygon an odd number of times, then the point $\mathbf{p}$ is inside the polygon.

Despite the above, problems remain where greater flexibility is required. Objects can contain areas where the texture variation might be considered noise. One thus wants to exclude such areas due to arguments similar to the above given. Another situation is that of having several structured objects, but in between those, the texture is highly unstructured. Features to accommodate such situations are implemented in the current AAM framework. Shapes are defined in terms of *paths*, which is a subset of the shape points with a given point connectivity. Each path can have a combination of the following properties:

- **Open/closed path** – Open path: a path where all points are connected to two points each.
- **Inner/outer path** – Outer path: a path where only the inside is included into the shape surface.
- **Original/artificial path** – Artificial path: a path added after the original annotation.
- **Hole/non-hole** – Hole: a path where the inside is excluded from the shape surface.

This provides a high degree of freedom, resulting in a more optimal AAM representation of the given problem. For further details, refer to [26].

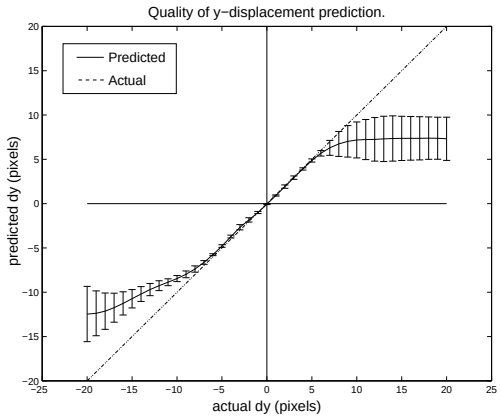



**Fig. 1**. Displacement plot for a series of model predications versus the actual displacement. Error bars are equal to one std.dev.

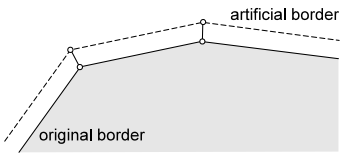
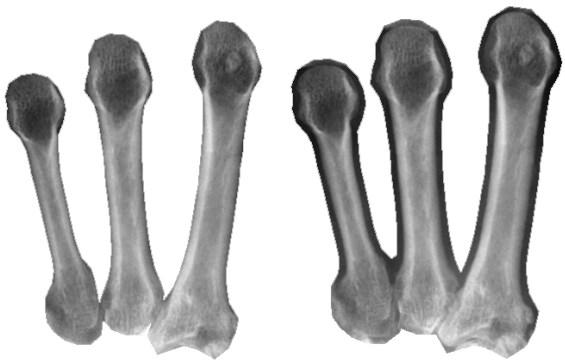



**Fig. 2**. Shape neighborhood added using an artificial border placed along the normal vectors of the original model points.



(a) (b)

**Fig. 3**. (a) Shape annotated using 150 landmarks. (b) Shape with a neighborhood region added resulting in $2 \times 150 = 300$ landmarks.

### 3.2. Neighborhood AAMs

While the removal of convex triangles gave greater shape control it also increases the risk of what we coin the *shrinking problem*. During matching of objects with a relatively homogeneous texture, matches sometimes tend to lie inside the real object. This is due to the fact that the AAMs evaluate the fit on the object texture only.

To avoid this we suggest including some neighboring region of the object. This will usually lead to more texture contrast in the model, since objects (often) are identified as something that stands out from its surroundings.

Neighborhood adding must be done carefully to avoid introducing new shape information. More precisely the shape PCA must retain its eigenvalue distribution. To accomplish this, we generate shape points fully correlated with the original points. The curvature is estimated at each original point. Then each new point is placed on the normal vector in a distance proportional to the relative size of the shape. See fig. 2. However, the texture PCA will suffer since the goal is to add new information, namely background pixels, of which one could expect a substantially higher degree of variation than across the object.

The metacarpal bones in fig. 3 serve as a good example of a shape with a relative homogeneous surface. By adding neighborhood the texture in fig. 3 (b) is substantially more specific than the shape without, fig. 3 (a).

### 3.3. Border AAMs

While the previous section provided a method for handling homogeneous objects, this section concerns the counter-example; heterogeneous objects.

For the described texture model, it is not possible to capture objects with large heterogeneity i.e. high texture variation. Think of this as a signal with a lack of structure. In such cases, we suggest to capture only an area around the outer rim, thereby excluding the "noisy" part of the object. This approach should be feasible since we (often) perceptually identify the outer rim due to it is structured behavior (often an abrupt change in intensity). We call this a border AAM. Using the enhanced shape representation, a border AAM is simply achieved by adding an interior path, which defines a hole and by adding an outer path as described in the previous section.

(a) (b)

**Fig. 4**. (a) Shape annotated using 83 landmarks. (b) Border shape with $3 \times 83 = 249$ landmarks.

By using this rationale AAMs can be made insensitive to large-scale texture noise inside the shapes, which otherwise would lead to a poor texture fit and a low landmark accuracy. The pork chops of fig. 4 constitute a good example of this situation, due to the heterogeneity of the complex structure of fat and meat from one training example to another.

To conclude this section, we stress that border AAMs also should be substantially faster than basic AAMs, since only a fraction of the original pixels is considered.

### 3.4. Fine-tuning the model fit

The AAM search provides a fast way of optimizing the AAM using prior knowledge. However, this might not lead to the optimal solution, primarily due to weakness in the assumption that the optimization problems in an AAM search are strictly similar to those observed in a training set. Thus, we suggest fine-tuning the model fit by using a general-purpose optimization method. This approach is feasible since it is reasonable to assume that we are close to the optimum after the traditional AAM search. Hence, the optimization fine-tuning should be possible in a reasonable amount of time. Though one is not guaranteed that the hyperspace is smooth at the position where the AAM search has converged, it is still more probable that we are near a well-behaved manifold in the hyperspace. The considered optimization methods are:

- **Gradient based methods:** Steepest descent, Conjugate gradient, Quasi-Newton (BFGS) [11]
- **Non-gradient based methods:** Pattern search [16]
- **Random-sampling based methods:** Simulated annealing [3, 21]

Preliminary experiments have shown that the random-sampling based method simulated annealing has best performance. Hence, this is the only optimization method considered in the experimental section. However, investigations are not conclusive since the performance of these methods is quite sensitive to configuration parameters – i.e. step sizes, standard deviations, stop criterions etc. Nevertheless, the decision is motivated by the observation that our objective function, $|\delta\mathbf{g}|^2$, is most likely non-convex. Thus, deterministic optimization techniques have a high risk of being caught in spurious minima's, whereas random-sampling techniques are more likely to escape.

### 3.5. Applying Robust Statistics

As seen earlier, AAM optimization is driven by texture differences, i.e. $|\delta\mathbf{g}|^2$. The measure, with which the optimization evaluates itself, is here forth named the *similarity measure*. However, the term *similar* is inherently vague in a mathematical sense. This section, will dwell on interpretations of the term similar that mimics the human ability of compensating for small numbers of gross errors, thus achieving robustness in recognition. These are called *robust* similarity measures where the term robust refers to the insensitivity to outliers. Cootes et al. [9] previously extended the basic AAM with learning-based matching to obtain robustness. This is achieved using a threshold for each element in $\delta\mathbf{g}$ estimated from the training set.

We suggest using robust similarity measures. To formalize the model fitting problem, a set of parameters, $\mathbf{c} = [c_1, \ldots, c_p]^{\mathrm{T}}$, are adjusted to fit a set of measurements (e.g. an image), $\mathbf{g} = [g_1, \ldots, g_m]^{\mathrm{T}}$. This is done by a minimization of the residuals:

$$E = \sum_{i=1}^{m} \rho(g_i - u(i, \mathbf{c}), \sigma_s) = \sum_{i=1}^{m} \rho(e_i, \sigma_s) \qquad (12)$$

where $u$ is a function that returns the model reconstruction of the $i^{\mathrm{th}}$ measurement and $\sigma_s$ is the scale parameter that determines what should be deemed outliers. The $\rho$-function determines the weighting of the residuals, and is also called the *error norm* where the most common error norm is the *quadratic norm*: $\rho(e_i) = e_i^2$. This is often referred to as the $L_2$ norm, which is the one used by basic AAMs, see (10).

It is easily seen, that the quadratic norm is notoriously sensitive to outliers, since these will contribute highly to the overall solution due the rapid growth of the $x^2$ function. It is therefore preferable to use a norm where the growth is more controlled. A smooth norm where the first derivative actually goes towards zero is the *Lorentzian estimator* [1]:

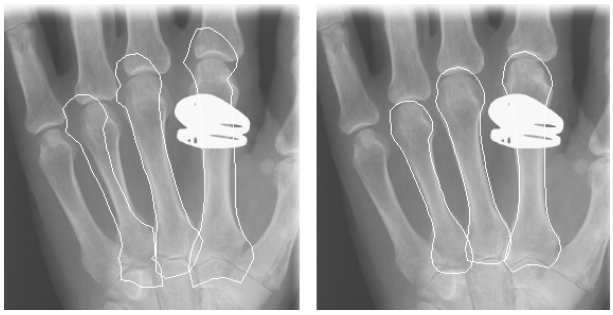


**Fig. 5**. Example of AAM search and Simulated Annealing fine-tuning, without (left) and with (right) the use of a robust similarity measure (Lorentzian error norm). Landmark error decreased from 7.0 to 2.4 pixels (pt.crv. error).

$$\rho(e_i, \sigma_s) = \log(1 + \frac{e_i^2}{2\sigma_s^2}) \qquad (13)$$

The Lorentzian norm has been integrated into the basic AAM to supplement the quadratic norm of Cootes et al. However, one should notice that even though the AAM-search evaluate its predictions using a robust measure, the predictions themselves are done using the pixel differences directly. To address this problem Cootes et al. [9] perform a thresholding of the texture vector before the prediction. This could be viewed upon as a robust preprocessing step. The threshold limit is estimated from the training set.

To demonstrate the effect of a robust error norm, an AAM search with fine-tuning using Simulated Annealing has been done *with* and *without* the Lorentzian estimator. Since radiographs are 2D projections of density, people wearing finger rings will have high-white outliers on one or more phalanges.[1] In the case given in fig. 5 the Lorentzian error norm was used. To simulate outliers the radiograph has been manipulated so that it appears as if the metacarpal is wearing a finger ring.[2] While not perfect, the robust AAM provides a substantially better fit compared to that of the basic AAM (both using simulated annealing).

For a further treatment of robust error norms and line processes in vision refer to [1].

### 3.6. Initialization

The basic AAM optimization scheme is inherently dependent on good initialization. To accommodate this, we devise the following search-based scheme thus making the use of AAMs fully automated. The technique is inspired by the work of Cootes et al. [9] who use a pixel difference evaluation criteria and a threshold estimation for detecting multiple object instances.

[1] Other outlier examples include highlights in perspective images and absence of interior parts, occlusion etc.

[2] Though this is highly unlikely since the metacarpals are situated in the middle of the hand.

The fact that the AAMs are self-contained is exploited in the initialization – i.e. they can fully synthesize (near) photo-realistic objects of the class that they represent concerning shape and textural appearance. Hence, we use the model without any additional data to perform the initialization.

The idea is to exploit an inherent property of the AAM-optimization – i.e. convergence within some range from the optimum. See e.g. fig. 1. This is utilized to narrow down an exhaustive search from a dense to a sparse population of the hyperspace spanned by pose- and **c**-parameters. In other words, normal AAM-optimizations are performed sparsely over the image using perturbations of the pose and model parameters.

This has proven to be both feasible, fast and robust. A set of relevant search configuration ranges is established and the sampling within this set is done as sparsely as possible. Any available prior knowledge about pose is utilized when determining search ranges.

The crucial part of this algorithm is somewhat inspired from the class of Genetic Algorithms.[3] The total set of search configurations constitutes the initial population of candidates. From this we let the $n$ fittest survive. These are then reproduced into more evolved guesses. From these the best is drawn and deemed the initial configuration. In pseudo-code, the initialization scheme for detecting one object per image is:

1. Set $m$ to a suitable low number (we use $m = 3$)
2. Establish a candidate set, $\{\mathbf{K}\}$, containing $n$ result entries
3. Obtain application specific search ranges within each parameter (e.g. $-\sigma_1 \leq c_1 \leq \sigma_1,\ x_{min} \leq x \leq x_{max}$, etc.)
4. Populate the space spanned by the ranges – as sparsely as the linear regression allows – by a set of search configurations $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$.
5. For each vector in $\mathbf{V}$
6. Do AAM optimization (max $m$ iterations)
7. Calculate the fit, $E = |\delta\mathbf{g}|^2$
8. If $E < \max_E \{\mathbf{K}\}$ add $(\mathbf{v}_i, E)$. If the number of elements in $\{\mathbf{K}\}$ exceeds $n$, then remove $\max_E \{\mathbf{K}\}$
9. End
10. For each element in $\{\mathbf{K}\}$
11. Do AAM optimization (max $k$ iterations, $k > m$)
12. Calculate and update the fit, $E = |\delta\mathbf{g}|^2$
13. End

The element in $\{\mathbf{K}\}$ with the minimum $E$ will now hold the initial configuration.

We stress that the application specific search ranges in step 3 are merely a help to increase initialization speed and robustness rather than a requirement. If no prior is known, step 3 is eliminated and an exhaustive search is performed.

[3]Notice however, while GAs are probabilistic, our technique is deterministic. Further, are the aspects of mutation and crossover in GAs not utilized here.

This scheme is readily extended into more than one object per image by a clustering of the candidate set using overlap tests. The approach in general can be accelerated substantially by searching in a multi-resolution (pyramidal) representation of the image. For a detailed treatment of initialization of deformable template models refer to [10].

## 4. EXPERIMENTAL RESULTS

Segmentation in medical images has always posed a difficult problem due to the special image modalities (CT, MRI, PET etc.) and the large biological variability. Thus, AAMs and the proposed extensions have been assessed on the three different modalities; see fig. 6 and below:

- **A – Radiographs of Metacarpals**
  Training set: 23 images (240×275 pixels)
  Shape model: 150 landmarks
  Texture model: $\sim$ 13.000 pixels
  95% variation explained using: 18 parameters
- **B – Cardiovascular MRIs**
  Training set: 13 images (256×191 pixels)
  Shape model: 83 landmarks
  Texture model: $\sim$ 15.000 pixels
  95% variation explained using: 10 parameters
- **C – Perspective images of Pork Chops**[4]
  Training set: 13 images (256×256 pixels)
  Shape model: 66 landmarks
  Texture model: $\sim$ 2.200 pixels
  95% variation explained using: 11 parameters

Common for all three cases is a relatively small training set (<30). Using larger training sets the chances of over constraining the model can be reduced, leading to more (representative) flexibility. However, since the actual construction of the training sets is done by manually placing landmarks on the images – a tedious and error prone task – one wants to keep the training sets relatively small. Methods for (semi)-automatic extraction of landmarks are reviewed in [26].

Each experiment consisted of building an AAM leaving one image, **I**, out. Subsequently the model was used to perform automatic segmentation in **I** using the proposed initialization scheme.[5] Each experiment was assed using the mean point to curve (point to associated border) measure and the mean intensity deviation measure [10]. Images were in grayscale 8-bit format. All figures in table 1 are means over all leave-one-out experiments. Failure was declared when the point to curve error exceeded 10 pixels. Performance-wise each MRI-optimization took 200 ms on average (PII 350 MHz).

[4]Due to controlled nature of these images (background is totally black), far simpler methods could have been used for segmentation. However, the large changes in texture and shape in these images lead to challenges suitable for evaluating the AAM approach. Therefore, from a strictly solution-oriented point of view, a cluttered background would have justified the use of AAMs more – and presumably achieved comparable accuracy.

[5]The sparse sampling in the initialization was done at a frequency of 12 pixels $x$ and $y$. **c**-parameters were kept constant zero.

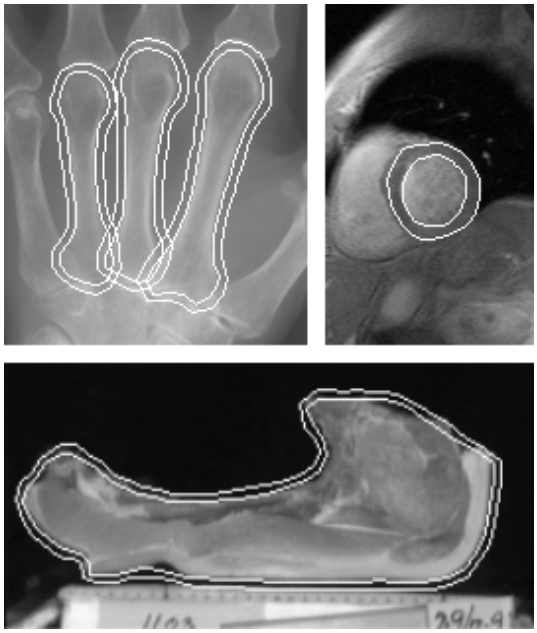


**Fig. 6**. Collage of the considered cases with successful segmentations. Upper left: A – Radiographs of Metacarpals. Upper right: B – Cardiovascular MRIs. Lower: C – Perspective images of Pork Chops. All images cropped to show details.

The minimum and maximum error for the tests with lowest average error in each case was:

- A 3: min. 0.53 / max. 1.01 pixels (point to curve)
- B 5: min. 0.60 / max. 1.34 pixels (point to curve)
- C 3: min. 0.65 / max. 2.43 pixels (point to curve)

In case A, the adding of neighborhood made the model more specific, removing the single failure in A1. Further fine-tuning using simulated annealing increased not only the explicit optimization criteria, but also the landmark accuracy. The neighborhood adding in case B also yielded higher landmark accuracy. Due to large-scale texture noise inside the object, the border AAM yielded the highest accuracy. Notice that the rather large texture error in B5 is not comparable to B1 – B4, since it is a completely different texture model. Contrary to case A, the cardiac AAM in case C possessed substantial structured contrast inside the object (the left ventricle), hence neighborhood-adding lead to a poorer fit. In all three cases, fine-tuning using simulated annealing improved both texture and landmark fit.

The lack of improvements using the robust Lorentzian similarity measure suggests that significant outliers as in fig. 5 were not present.

## 5. FUTURE WORK

We are currently investigating several methods, which extend the AAM scheme. Flexibility of the shapes are being enhanced by unifying Finite Element Models and AAMs by adding artificial interior points, which are deformed by an FEM. Further an Active Texture Weighting scheme is being designed which will add more flexibility to the texture model representation. Each texture pixel is given a weight, which is determined 1) manually, by drawing a semi-transparent mask or 2) automatically, by some function of the pixel variance over the training set. In the latter case, we expect this automatic method to supersede the manual decision of using a normal, neighborhood or border AAM. AAMs also extend to higher spatial dimensions [9], which will be the topic of our long-term future research.

**Table 1**. Leave-one-out test results for case studies.

| # | Type | Point to curve deviation (pixels) | Mean intensity deviation | Init. fail. |
|---|---|---|---|---|
| **A – Metacarpals** | | | | |
| 1 | Basic AAM | 0.88 | 4.9 | 1 |
| 2 | 1+Neighborhood | 0.84 | 5.2 | 0 |
| 3 | 2+SA | 0.82 | 5.0 | 0 |
| 4 | 3+Lorentzian | 0.83 | 5.0 | 0 |
| **B – Pork Chops** | | | | |
| 1 | Basic AAM | 1.12 | 13.2 | 0 |
| 2 | 1+Neighborhood | 0.91 | 13.9 | 0 |
| 3 | 2+SA | 0.89 | 13.6 | 0 |
| 4 | 3+Lorentzian | 0.91 | 13.6 | 0 |
| 5 | Border AAM | 0.86 | 23.5 | 0 |
| **C – Cardiac MRIs** | | | | |
| 1 | Basic AAM | 1.18 | 7.1 | 0 |
| 2 | 1+Neighborhood | 1.73 | 7.5 | 0 |
| 3 | 1+SA | 1.06 | 5.9 | 0 |
| 4 | 3+Lorentzian | 1.13 | 6.0 | 0 |

## 6. IMPLEMENTATION

All experiments, illustrations etc. have been made using our Active Appearance Models Application Programmers Interface (AAM-API) developed in the C++ language. The API is released under the open source initiative, which means that others freely can download, use and elaborate on the AAM-API. Effort has been put into providing documentation and educational features such as movie generation of the modes of variation, model search etc. Further information on AAMs, source code and documentation can be obtained at http://www.imm.dtu.dk/∼aam/.

## 7. CONCLUSION

In this paper, we have presented a set of extensions which all yield higher landmark accuracy when applied to the type of situations they address. The proposed extensions are an enhanced representation of shape used for handling of homogeneous and heterogeneous textures and a fine-tuning of the AAM optimization with and without the usage of robust error norms. Finally, the usage of AAMs is fully automated by the presented initialization scheme.

The performance has been assessed on three different image modalities – i.e. radiographs, perspective images and

magnetic resonance images, reaching a mean landmark location accuracy of 0.82, 0.86 and 1.06 pixels (pt.crv.), respectively, all using a relatively small training set of 23, 13 and 13 examples. The experiments were accomplished with no manual interaction. The implementation was unchanged in all three cases. No parameters were adjusted to produce the results.[6]

The three cases stress the fact that the AAM approach with the proposed extensions is a fully automated, general vision technique that captures domain knowledge through observation. Furthermore, we have experienced the AAM approach with the proposed extensions to be data-driven, self-contained and fast.

## 8. ACKNOWLEDGEMENTS

Cardiac MRIs were provided and annotated by M.D. Jens Christian Nilsson and M.D. Bjørn A. Grønning, Danish Research Center of Magnetic Resonance, H:S Hvidovre Hospital. M.Sc.Eng. Torben Lund provided the annotation tool. M.D. Lars Hyldstrup, Department of Endocrinology, H:S Hvidovre Hospital provided, and Pronosco A/S digitized, the metacarpal radiographs. The work of M. B. Stegmann was partly supported by a grant from Pronosco A/S.

## 9. REFERENCES

[1] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. Journal of Computer Vision*, 19(1):57–92, 1996.

[2] F. L. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–85, 1989.

[3] V. Cerny. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Jour. of Optimization Theory and Applications*, 45:41–51, 1985.

[4] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proc. European Conf. on Computer Vision*, volume 2, pages 484–498. Springer, 1998.

[5] T. F. Cootes and C. J. Taylor. Combining elastic and statistical models of appearance variation. In *Proc. European Conf. on Computer Vision*, volume 1, pages 149–163, 2000.

[6] T. F. Cootes and C. J Taylor. *Statistical Models of Appearance for Computer Vision*. Tech. Report , University of Manchester, http://www.isbe.man.ac.uk/∼bim/, Feb. 2000.

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

[8] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. John Wiley & Sons, 1998.

[9] G. J. Edwards, T.F. Cootes, and C. J. Taylor. Advances in active appearance models. In *Proc. Int. Conf. on Computer Vision*, pages 137–142, 1999.

[6] To the benefit of future research activities and comparative studies, we intend to make all images and annotations available at our website.

[10] R. Fisker. *Making Deformable Template Models Operational*. PhD thesis, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, 2000. http://www.imm.dtu.dk/documents/ftp/phdliste/phdliste.html.

[11] R. Fletcher. *Practical Methods of Optimization*. John Wiley & Sons, 1987.

[12] C. A. Glasbey and K. V. Mardia. A review of image-warping methods. *Journal of Applied Statistics*, 25(2):155–172, 1998.

[13] M. Gleicher. Projective registration with difference decomposition. In *Proc. 1997 Conf. on Computer Vision and Pattern Recognition*, pages 331–337. IEEE Comput. Soc, 1997.

[14] C. Goodall. Procrustes methods in the statistical analysis of shape. *Jour. Royal Statistical Society, Series B*, 53:285–339, 1991.

[15] U. Grenander, Y. Chow, and D. M. Keenan. *Hands: A Pattern Theoretic Study of Biological Shapes*. Springer, 1991.

[16] R. Hooke and T. A. Jeeves. Direct search: solution of numerical and statistical problems. *Jour. Assoc. Comput.*, 8(212-229), 1961.

[17] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[18] A. K. Jain, Y. Zhong, and M.-P. Dubuisson-Jolly. Deformable template models: A review. *Signal Processing*, 71(2):109–129, 1998.

[19] A. K. Jain, Y. Zhong, and S. Lakshmanan. Object matching using deformable templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(3):267–278, 1996.

[20] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. Jour. of Computer Vision*, 8(2):321–331, 1988.

[21] S. Kirkpatrick, C. D. Gellant, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[22] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 2(1):91–108, 1996.

[23] 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*, volume 1. SPIE, 2000.

[24] S. Sclaroff and J. Isidoro. Active blobs. *Proc. of the Int. Conf. on Comput. Vision*, pages 1146–1153, 1998.

[25] J.R. Shewchuk. Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In *Applied Computational Geometry. FCRC'96 Workshop.*, pages 203–222. Springer-Verlag, 1996.

[26] M. B. Stegmann. Active appearance models: Theory, extensions and cases. Master's thesis, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, 2000. http://www.imm.dtu.dk/∼aam/.

[27] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proc. 1991 IEEE Com. Soc. Conf. on CVPR*, pages 586–91. IEEE Com. Soc. Press, 1991.

[28] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *Int. Jour. of Computer Vision*, 8(2):99–111, 1992.