Active Appearance Models: Theory and Cases

M. B. Stegmann[1]*, R. Fisker[1], B. K. Ersbøll[1], H. H. Thodberg[2], L. Hyldstrup[3]

[1]Department of Mathematical Modelling
Technical University of Denmark
DTU Building 321, DK-2800 Lyngby, Denmark

[2]Pronosco A/S, Kohavevej 5, DK-2950 Vedbæk, Denmark
[3]H:S Hvidovre Hospital, Kettegård Allé 30, DK-2650 Hvidovre, Denmark

Abstract

In this paper, we present a general approach towards image segmentation using the deformable model Active Appearance Model (AAM) as proposed by Cootes et al. A priori knowledge is learned through observation of shape and texture variation in a training set and is used to obtain a compact object class description, which can be used to rapidly search images for new object instances. An overview of the theory behind AAMs is given followed by an improved initialization scheme, thus making the AAMs fully automated. Finally, two cases are presented. It is demonstrated that AAMs can successfully segment bone structures in radiographs of human hands and structures of the human heart in 2D extracts of 4D cardiovascular magnetic resonance images. The observed mean point location accuracy was 1.0 and 1.3 pixels, respectively.

Keywords: Deformable Models, Snakes, Principal Component Analysis, Shape Analysis, Non-Rigid Object Segmentation, Initialization, Metacarpal Radiographs, Cardiovascular Magnetic Resonance Imaging.

1 Introduction

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

Among the earliest and most well known deformable models is the Active Contour Model – known as *Snakes* proposed by Kass et al. [17]. 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 model have been carried out. These include Yuille's et al. [25] parameterization of a human eye using ellipsis and arcs.

In a more general approach, while preserving specificity Cootes et al. [7] proposed the Active Shape Models (ASM) where shape variability is learned through 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 [3]. Besides shape information, the textual information, i.e. the pixel intensities across the object, is included into the model. AAMs has been further developed in [5, 9, 6].

Quite similar to AAMs and developed in parallel herewith, Sclaroff & Isidoro suggested the Active Blob approach [20, 15]. Active Blobs is a real-time tracking technique, which captures shape and textual information from a prototype image using a finite element model (FEM) to model shape variation. Compared to AAMs, Active Blobs deform a static texture, whereas AAMs change both texture and shape during the optimization.

For further information on deformable models, refer to the surveys given in [16, 18].

2 Active Appearance Models

Below we describe the outline of the Active Appearance Model approach. AAMs distinguish themselves in the sense that segmentation can be carried out using the approach as a black box. We need only provide with domain knowledge in the form of a training set annotated by specialists (e.g. radiologists etc.).

Described is the training of the model, the modelling

*Corresponding author: aam@imm.dtu.dk

of shape and texture variation and the optimization of the model. Finally, an improved method for automated initialization of AAMs is devised.

For a commented pictorial elaboration on the sections below – including the alignment process – refer to appendix A.

2.1 Shape & Landmarks

The first matter to clarify is: What do we actually understand by the term *shape*? This paper will adapt the definition by D.G. Kendall [8]:

> **Definition 1**: **Shape** *is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object.*

The next question that naturally arises is: How should one describe a shape? In everyday conversation unknown shapes are often described by referring to well-known shapes – e.g. *"Italy has the shape of a boot"*. Such descriptions can obviously not be utilized in an algorithmic framework.

One way of representing shape is by locating a finite number of points on the outline. Consequently the concept of a *landmark* is adapted [8]:

> **Definition 2**: *A* **landmark** *is a point of correspondence on each object that matches between and within populations.*

A mathematical representation of an n-point shape in k dimensions could be concatenating each dimension in a kn-vector. The vector representation for planar shapes would then be:

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

Notice that the above representation does not contain any explicit information about the point connectivity.

2.2 Shape Formulation

A classical statistical method for dealing with redundancy in multivariate data – such as shapes – is 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 data point in a $2n^{\mathrm{th}}$ 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. the shape analysis is performed on the true shapes according to the definition. As shape metric in the alignment the Procrustes distance [13] is used. Other shape metrics such as the Hausdorff distance [14] 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:

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \tag{2}$$

The maximum likelihood estimate of the covariance matrix can thus be given as:

$$\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^{\mathrm{T}} \tag{3}$$

The principal axis of the $2n^{th}$ dimensional shape ellipsoid are now given as the eigenvectors, $\mathbf{\Phi}_s$, of the covariance matrix.

$$\mathbf{\Sigma\Phi}_s = \mathbf{\Phi}_s\mathbf{\lambda}_s \tag{4}$$

A new shape instance can then be generated by deforming the mean shape by a linear combination of eigenvectors, weighted by $\mathbf{b}_s$, also called the modal deformation parameters.

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{\Phi}_s\mathbf{b}_s \tag{5}$$

Essentially, the point or *nodal representation* of shape has now been transformed into a *modal representation* where modes are ordered according to their *deformation energy* – i.e. the percentage of variation that they explain.

What remains is to determine how many modes to retain. This leads to a trade off between the accuracy and the compactness of the model. However, it is safe to consider small-scale variation as noise. It can be shown that the variance across the axis corresponding to the i^{th} eigenvalue equals the eigenvalue itself, λ_i. Thus to retain p percent of the variation in the training set, t modes can be chosen satisfying:

$$\sum_{i=1}^{t}\lambda_i \geq \frac{p}{100}\sum_{i=1}^{2n}\lambda_i \tag{6}$$

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. The main reason for this is that most literature on AAMs uses this definition of texture, probably due to the

close resemblance of some of the AAM techniques to techniques in computer graphics.

In computer graphics, the term texture relates directly to the pixels mapped upon virtual 2D and 3D surfaces. Thus, we derive the following definition:

> **Definition 3: Texture** *is 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:

$$\mathbf{g} = (g_1, \ldots, g_m)^{\mathrm{T}} \tag{7}$$

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 warping function needs to be established. This can be done in several ways. Here, a piece-wise affine warp based on the Delaunay triangulation of the mean shape is used. Another, theoretically better, approach might be to use thin-plate splines as proposed by Bookstein [1]. For details on the Delaunay triangulation and image warping refer to [11, 21].

Following the warp sampling of pixels, a photometric normalization of the $\mathbf{g}$-vectors of the training set is done 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:

$$\mathbf{g} = \overline{\mathbf{g}} + \mathbf{\Phi}_g \mathbf{b}_g \tag{8}$$

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.

Notice that there will always be far more dimensions in the samples than observations 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. Consult [5, 22] or a textbook in statistics for the details.

2.3.1 Combined Model Formulation

To remove correlation between shape and texture model parameters – and to make the model representation 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, $\mathbf{c}$:

$$\mathbf{b} = \mathbf{Q}\mathbf{c} \tag{9}$$

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

$$\mathbf{b} = \begin{pmatrix} \mathbf{W}_s \mathbf{b}_s \\ \mathbf{b}_g \end{pmatrix} = \begin{pmatrix} \mathbf{W}_s \mathbf{\Phi}_s^{\mathrm{T}} (\mathbf{x} - \overline{\mathbf{x}}) \\ \mathbf{\Phi}_g^{\mathrm{T}} (\mathbf{g} - \overline{\mathbf{g}}) \end{pmatrix} \tag{10}$$

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

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

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{\Phi}_s \mathbf{W}_s^{-1} \mathbf{Q}_s \mathbf{c} \tag{11}$$

$$\mathbf{g} = \overline{\mathbf{g}} + \mathbf{\Phi}_g \mathbf{Q}_g \mathbf{c} \tag{12}$$

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.

Observe that 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.

2.4 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.

In this way by adjusting the AAM-parameters ($\mathbf{c}$ and pose) the model can deform to fit the image in the best possible way.

Though we have seen that the parameterization of the object class in question can be compacted markedly by the principal component analysis it is far from an easy task to optimize the system. This is not only computationally cumbersome but also theoretically challenging – optimization theory-wise – since it is not guaranteed that the search-hyperspace is smooth and convex.

However, AAMs circumvent these potential problems in a rather untraditional fashion. The key observation is that each model search constitutes what we call a *prototype search* – the search path and the optimal model parameters are unique in each search where the final model configuration matches this configuration.

These prototype searches can be performed at model building time; thus saving the computationally expensive high-dimensional optimization. Below is described how to collect these prototype searches and how to utilizes them into a run-time efficient model search of an image.

It should be noticed that the Active Blobs approach is optimized using a method quite similar to that of AAMs named *difference decomposition* as introduced by Gleicher [12].

2.4.1 Solving Parameter Optimization Offline

It is proposed that the spatial pattern in $\delta\mathbf{I}$ can predict the needed adjustments in the model and pose parameters to minimize the difference between the synthesized object delivered by the AAM and an actual image, $\delta\mathbf{I}$:

$$\delta\mathbf{I} = \mathbf{I}_{image} - \mathbf{I}_{model} \tag{13}$$

The simplest model we can arrive at constitutes a linear relationship:

$$\delta\mathbf{c} = \mathbf{R}\delta\mathbf{I} \tag{14}$$

To determine a suitable $\mathbf{R}$ in equation (14), a set of experiments are conducted, the results of which are fed into a multivariate linear regression using principal component regression due to the dimensionality of the texture vectors. Each experiment displaces the parameters in question by a known amount and measuring the difference between the model and the image-part covered by the model.

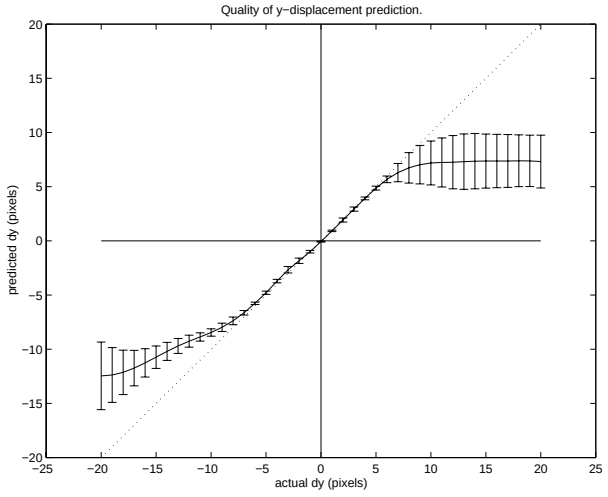

Figure 1: Displacement plot for a series of y-pose parameter displacements. Actual displacement versus model prediction. Error bars are 1 std.dev.

As evaluation of the assumption of a linear relationship between the model and pose parameters and the observed texture differences, figure 1 shows the actual and the mean predicted displacement from a number of displacements. The error bars correspond to one standard deviation.

Hence, the optimization is performed as a set of iterations, where the linear model, in each iteration, predicts a set of changes in the pose and model parameters leading to a better model to image fit. Convergence is declared when an error measure is below a suitable threshold.

As error measure, we use the squared L_2 norm of the texture difference, $|\delta\mathbf{g}|^2$. To gain a higher degree of robustness, one might consider using the Mahalanobis distance or a robust norm such as the Lorentzian error norm [20]. Fitness functions allowing for global non-linear transformations such as the mutual information [23, 24] measure might also be considered.

2.5 Initialization

The optimization scheme described above is inherently sensitive to a good initialization. To accommodate this, we devise the following search-based scheme thus making the use of AAMs fully automated. The technique is somewhat inspired by the work of Cootes et al. [9].

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

The idea is to use the inherent properties of the AAM-optimization – i.e. convergence within some range from the optimum. This is utilized to narrow an exhaustive search from a dense to sparse population of the hyperspace spanned by pose- and $\mathbf{c}$-parameters. In other words, normal AAM-optimizations are performed sparsely over the image using perturbations of the model parameters.

This has proven to be both feasible and robust. A set of relevant search configuration ranges is established and the sampling within this is done as sparsely as possible.

Consider the graph given in figure 1, which demonstrates that it should be safe to sample the y-parameter with a frequency of at least 10 pixels. One could also claim that as long the prediction preserves the right sign it is only a matter of sufficient iteration.

To achieve sensitivity to pixel outliers we use the variance of the square difference vector between the model and the image as error measure:

$$e_{fit} = V[\delta\mathbf{g}^2] \tag{15}$$

As in the optimization this could easily be improved

by using more elaborate error measures. In pseudo-code, the initialization scheme for detecting one object per image is:

1. Set $e_{min} = \infty$ and m to a suitable low number (we use $m = 3$)
2. Obtain application specific search ranges within each parameter (e.g. $-\sigma \leq c_1 \leq \sigma$ etc.)
3. Populate the space spanned by the ranges – as sparsely as the linear regression allows – by a set of sampling vectors $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$.
4. For each vector in $\mathbf{V}$
5. Do AAM optimization (max m iterations)
6. Calulate the fit, e_{fit}, as given by (15)
7. If $e_{fit} < e_{min}$ Then $e_{min} = e_{fit}$, $\mathbf{v}_{fit} = \mathbf{v}_n$
8. End

The vector $\mathbf{v}_{fit}$ will now hold the initial configuration.

Notice that the application specific search ranges in step 2 is merely a help to increase initialization time and robustness than it is a requirement. If nothing is known beforehand, step 2 is eliminated and an exhaustive search is performed.

This approach can be accelerated substantially by searching in a multi-resolution (pyramidal) representation of the image.

3 Implementation

All experiments, illustrations etc. have been made using the Active Appearance Models Application Programmers Interface (AAM-API) developed in the C++ language and based on the Windows NT platform. This API will be released under the open source initiative in Autumn 2000[1], which means that other freely can download, use and elaborate on the AAM-API.

The foundation for the AAM-API is the Intel Math Kernel Library for fast MMX implementation of BLAS, MS VisionSDK for image handling, ImageMagick for image I/O and finally DIVA for image processing and matrix handling.

AAM-API performance compares to that of Cootes et al. As an example, the MRI optimizations took each 200 ms on average on a PII 350 MHz. Much effort has been put into providing documentation and educational features such as movies of the modes of variation, model search etc.

Further info on AAMs, the AAM-API and full source code documentation can be obtained at the AAM-site.[2]

[1] Probably under the GNU Public License.

[2] `http://www.imm.dtu.dk/~aam/`

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. To assess the performance of AAMs in such environments, our implementation has been tested upon radiographs of human hands and cardiac MRIs.

4.1 Radiographs of Metacarpals

Segmentation in radiographs (x-rays) pose a difficult problem due to large shape variability and inherent ambiguity of radiographs. This forms a suitable challenge. Other attempts to perform segmentation in radiographs include the work of Efford [10], where ASMs and other methods were used.

Twenty radiographs of the human hand were obtained and three metacarpal bones were annotated using 50 points on each. The annotation of metacarpals 2,3 and 4 were concatenated into a 150-point model. To incorporate a more substantial texture contrast into the model, additional 150 points were placed along the normal of each model point, thus arriving at a 300-point shape model. The texture model consisted of approx. 10.000 pixels. Using 16 model parameters, 95% of the variation in the training set were explained.

Automated initialization of the model followed by optimization reached a mean location accuracy of 0.98 pixels (point to associated border [2, 4]) when testing on four unseen images with ground truth annotations. The mean texture error was approx. 7 gray levels (input images were in the byte range).

Examples of initialization and optimization are given in the figures 2-5. Notice a fairly good fit even in the distal (upper) and proximal (lower) end of the metacarpals where radiographs are rather ambiguous.

To asses the performance *within* points, the mean point to point distance is plotted in figure 6. Not surprisingly, problems arise in the distal and proximal end of the metacarpals due the large shape variability and the ambiguous nature of radiographs in regions of overlap.

4.2 Cardiac MRIs

Another application in medical imaging is locating structures of 4D (space, time) cardiovascular magnetic resonance images. Temporal registration relative to the heart cycle has been done using ECG-triggered image acquisition. The pixel depth was 8 bits.

An AAM has been built upon only four – spatially and temporally corresponding – 2D slices of four different hearts. The endocardial and epicardial con-

Figure 2: Model border after automated initialization (cropped).

Figure 3: Optimized model border.

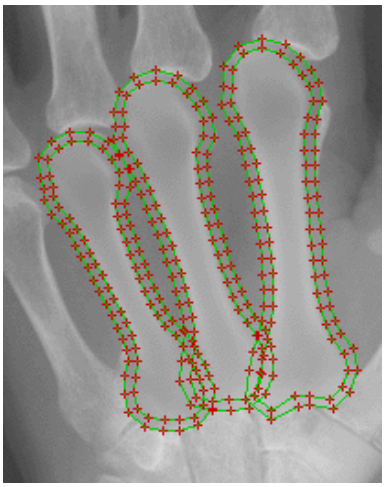

Figure 4: AAM after automated initialization (cropped).

tour have been annotated using 30 points each. The resolution of the 2D slices was 256x256 pixels resulting in a texture model of approx. 2600 pixels. The combined PCA explains 84 % of the variation in the

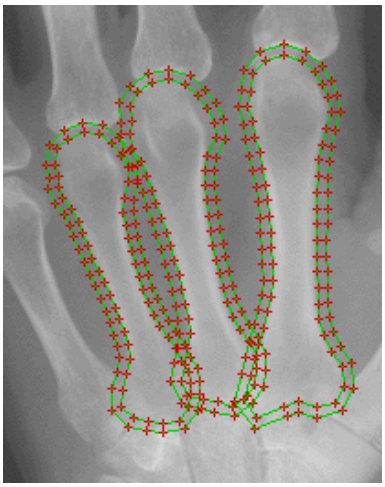

Figure 5: Optimized AAM (cropped).

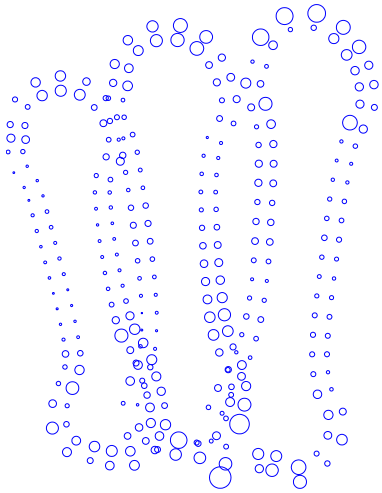

Figure 6: Mean point to point deviation from the ground truth annotation of each metacarpal. Low location accuracy is observed at the distal and proximal ends.

training set using two model parameters.

The cardiac AAM was then used to search in a unseen image spatially and temporally similar to those in the training set. The described initialization technique reached the result seen in figures 7 and 9. A final optimization reached a mean point accuracy of 1.7 pixels (point to associated border). The result can be seen in figures 8 and 10. The mean texture error was approx. 11 gray levels. By incorporating a model neighborhood similar to the metacarpal model the mean point accuracy was increased to 1.3 pixels.

Future work on the cardiac AAM will include models built upon extracts differing spatially and temporally, thus leading to a somewhat unified AAM of a larger subspace from the original 4D. For a commented pictorial of the cardiac AAM, refer to appendix A.

AAM segmentation of 2D cardiac MRIs has previ-

ously been done by Mitchell et al. [19]. A total of 102 images were used for the training set reaching a mean point accuracy of approx. 1 pixel on the endocardial and epicardial contour. Annotated structures were the right ventricle and endocardial and epicardial contours. The model was initialized manually.

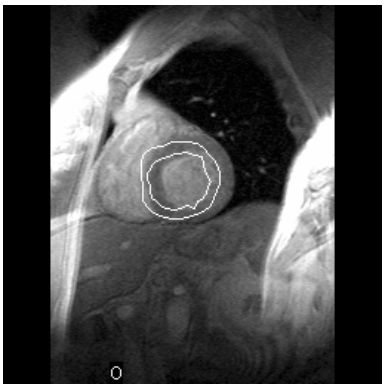

Figure 7: Model border after automated initialization.

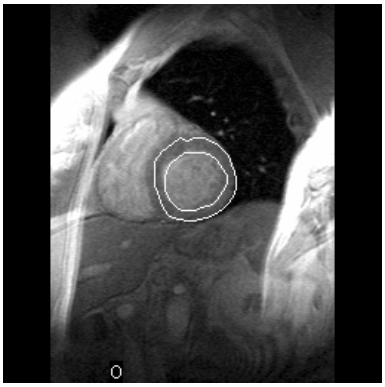

Figure 8: Optimized model border.

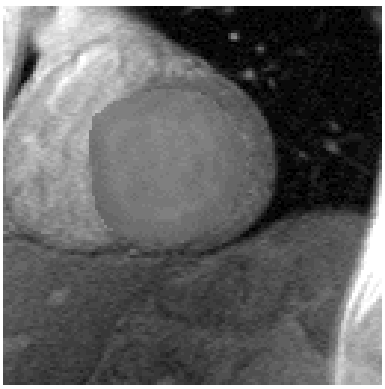

Figure 9: AAM after automated initialization (cropped).

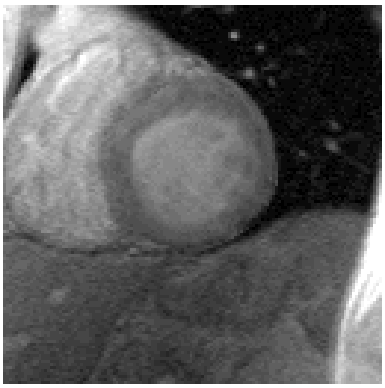

Figure 10: Optimized AAM (cropped).

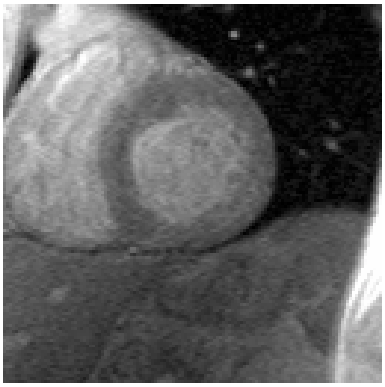

Figure 11: Original image (cropped).

5 Discussion & Conclusions

In this paper we have presented the basic theory of AAMs and devised a method providing sufficient initialization of AAMs.

The performance of AAMs has been assessed on two different image modalities - i.e. radiographs and magnetic resonance images reaching a mean point location accuracy of 1.0 and 1.3 pixels, respectively. In both cases the location accuracy was noteworthy increased by adding a suitable neighborhood to the outer contours of the model, thus enhancing textual contrast.

In the MRI case, we have shown that even with a training set as small as four examples, very good segmentation results can be obtained. This leads towards the straightforward assumption that the less variation observed in the object class in question, the smaller the training set one can allow. However – more than four images would probably still be desirable.

The two cases stress the fact that the AAM approach is a example of a general vision technique that capture domain knowledge through observation. Contrary to this the bulk part of model-based vision techniques has hand crafted a priori knowledge by design and

programming. Furthermore, we have experienced the AAM approach to be data-driven (non-parametric), self-contained and fast. We also notice that AAMs extend to multivariate imaging and higher spatial dimensions - i.e. into 3D models etc.

More information on AAMs, papers, presentations, movies of model searches etc. can be obtained at the AAM-site at `http://www.imm.dtu.dk/~aam/`. A more comprehensive treatment of the work of M. B. Stegmann can be found in [22].

6 Acknowledgements

Cardiac MRIs were provided and annotated by M.D. Jens Christian Nilsson and M.D. Bjørn A. Grønning, H:S Hvidovre Hospital. M.Sc.Eng. Torben Lund provided the annotation tool.

Dr. Rasmus Larsen, M.D. Jens Christian Nilsson and M.D. Bjørn A. Grønning is acknowledged for their very useful comments on the manuscript.

The work of M. B. Stegmann is in part supported by a grant from Pronosco A/S.

References

[1] 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.

[2] T. F. Cootes, G. Edwards, and C. J. Taylor. A comparative evaluation of active appearance model algorithms. In *BMVC 98. Proc.of the Ninth British Machine Vision Conf.*, volume 2, pages 680–689. Univ. Southampton, 1998.

[3] 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.

[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.*, pages 173–182, 1999.

[5] T. F. Cootes and C. J Taylor. *Statistical Models of Appearance for Computer Vision*. Tech. Report, ed. 02-03-2000 , University of Manchester, http://www.isbe.man.ac.uk/~bim/, 1999.

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

[10] N. D. Efford. *Knowledge-Based Segmentation and Feature analysis of Hand Wrist Radiographs*. Tech. Report, University of Leeds, 1994.

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

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

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

[14] 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.

[15] J. Isidoro and S. Sclaroff. Active voodoo dolls: a vision based input device for nonrigid control. In *Proc. Computer Animation '98*, pages 137–143. IEEE Comput. Soc, 1998.

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

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

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

[19] 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.

[20] S. Sclaroff and J. Isidoro. Active blobs. *Proc. of the IEEE ICCV*, pages 1146–1153, 1998.

[21] 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.

[22] M. B. Stegmann. Active appearance models: Theory and cases. Master's thesis, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, 2000.

[23] C. Studholme, D. L. G. Hill, and D. J. Hawkes. An overlap invariant entropy measure of 3D medical image alignment. *Pattern Recognition*, 32(1):71–86, 1999.

[24] P. Viola and W. M. Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

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

A Illustrated Cardiac AAM

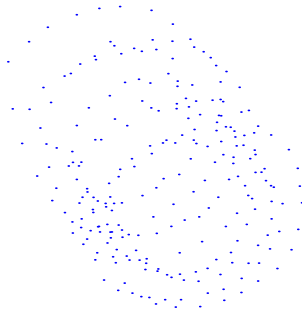

Figure 12: Point cloud of four unaligned heart chamber annotations.

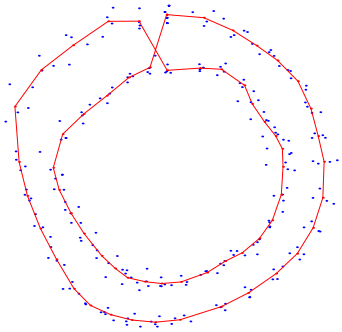

Figure 13: Point cloud of four aligned heart chamber annotations with mean shape fully drawn.

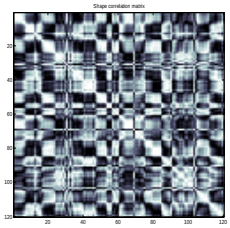

Figure 14: Correlation matrix of the four annotations. Observe the obvious point correlations.

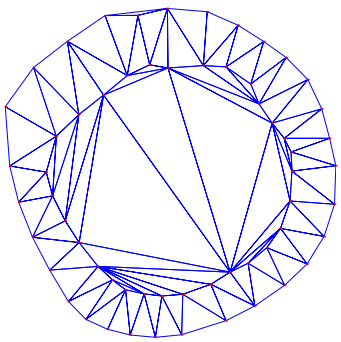

Figure 15: Delanay triangulation of the mean shape.

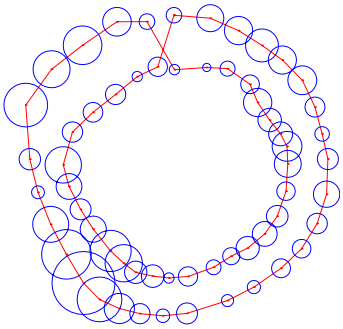

Figure 16: Point variation of the four annotations; radius $= \sigma_x + \sigma_y$. Notice the large point variation to the lower left.

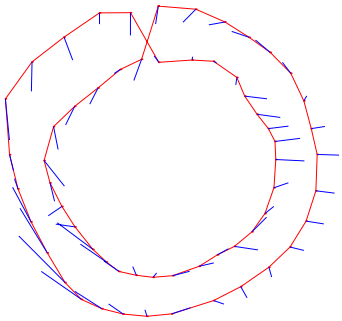

Figure 17: The first eigenvector plotted as displacement vectors. Notice that the large point variation observed in figure 16 is point variation along the contour, which only contributes to a less compact model contrary to explaining actual shape variation.

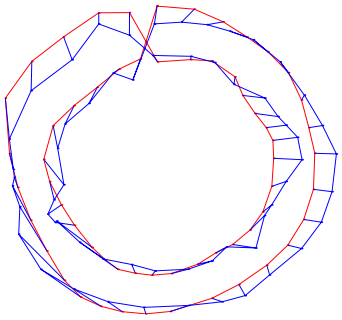

Figure 18: Mean shape and shape deformed by the first eigenvector. Notice that this emphasizes the point above; that a lot of the deformation energy does not contribute to any actual shape changes.