Active Shape Models for Face Detection

John W. Miller

***Abstract*—The ability to detect and segment objects within an image is useful across all types of image analysis. Model-based vision relies on statistical information collected from training images to search for object features in new images. Introduced in 1995 by Cootes et al. [1], active shape models (ASMs) is a model-based approach that at the time became a cutting-edge technique for object detection and segmentation. This paper motivates the need for automatic detection techniques, covers the mathematical techniques behind ASMs, and demonstrates an implementation of ASMs for face detection. When evaluated on a test set of face images, the ASM implementation presented here was able to accurately converge on facial features in nearly every image.**

***Keywords*—Active shape models, face detection, point distribution models, Procrustes analysis, principal component analysis.**

I. Introduction

Object detection and segmentation is a crucial tool across all areas of image analysis. In medical applications, segmentation is used to automatically quantify brain tumors, detect shadows in lung X-rays, and distinguish layers of the retina, among many other uses. Segmentation techniques that are fast, accurate, and robust continue to be large focus in image analysis research.

Techniques for segmentation vary from graph-based approaches, to deep learning algorithms, to modified registration techniques. Active shape models (ASMs) introduced by Cootes et al. in 1995 [1] is an approach for object identification and segmentation that employs training data to intelligently deform a contour around an object in an image. The technique was in part, introduced to address the shortcomings of active contour models, which deform with no awareness of the object characteristics.

Object detection has many applications outside of medical imaging as well. With the prevalence of digital cameras and mass collections of social images, face detection has become increasingly popular. Face detection was chosen for this paper to test the capabilities of ASMs using an example with popular appeal.

This paper will cover the mathematical techniques behind active shape models, presents the results of ASMs trained to search for faces, and discusses the shortcomings of and possible improvements to ASMs. Additionally, an example of a potential novel application of ASMs to medical imaging will be shown.

J. Miller is with the Department of Electrical and Computer Engineering, University of Iowa, Iowa City, IA, 52246 USA e-mail: john-w-miller@uiowa.edu.

II. Methods

The high-level pipeline for this ASM implementation is as follows:

1) Manually label a set of training images
2) Align the landmarks within the training set
3) Create a statistical shape model from the aligned shapes
4) Create another statistical model from the gray-level pixel values at each landmark point
5) Use the two models to iteratively search for an object within a new image

The mathematics underlying this ASM implementation are fairly straightforward. Principal components analysis (PCA) is used to create both the shape and gray-level models, and Procrustes analysis is used to align the shapes. Both will be described in detail below. In general, this paper follows the description and notation from [1].

A. Shape representation

An object's shape is represented as a vector $\mathbf{x}$ of length $2n$:

$$\mathbf{x} = (x_0, y_0, x_1, y_1, \ldots, x_{n-1}, y_{n-1})^T \tag{1}$$

where n is the number of landmark points assigned to the shape. A key requirement of ASMs is that the objects being detected share consistent features from image to image. For example, most images of faces will contain consistent features such as the edges of eyes, nostrils, or lips that are easy to identify and designate as landmarks across different examples. Conversely, an images of brain tumors, for example, may not share consistent features, making it difficult to determine where to place landmarks. The ASM technique assumes both shape and gray-level information will be relatively consistent at each landmark across each image.The ASM described in this paper was built from 50 manually-labeled grayscale images of faces from [3]. An example of a face from the training set labeled by the author is shown in Fig. 1.

B. Procrustes analysis for shape alignment

The shapes in a training set likely vary in translation, rotation, and scale between the various training images. Before the manually-labeled shapes can be used to create a statistical model, this variation should be accounted for by aligning the shapes. If the shapes are not aligned, and the training images vary considerably, the model may not be comparing equivalent points between images. However, in certain applications (as was the case in this implementation) it can be beneficial to leave the scaling portion out of the alignment procedure. By keeping variations in scale in the training set, the model will learn to deform itself to account for changes in scale in new images. Often the first principal component (PC) of the PCA

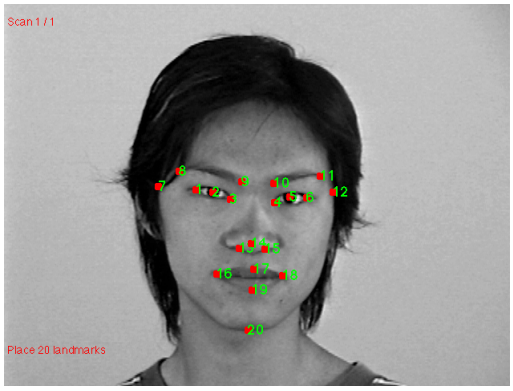

Fig. 1. A face from the training set, labeled by the author.

model will account for scale if scale information is left in the training images, and the other PCs will account for variations specific to the object. Additionally, aligning the shapes can help make their distribution more Gaussian, an underlying assumption of the PCA model.

The shapes are aligned using a method known as Procrustes analysis. Procrustes analysis attempts to find the rotation, θ, scale, s, and translation, (t_x, t_y), parameters that optimally align two or more shapes. The optimal parameters are those that minimize the weighted sum

$$\mathbf{E}_j = (\mathbf{x}_i - \mathbf{M}(s_j, \theta_j)[\mathbf{x}_j] - \mathbf{t}_j)^T \mathbf{W} (\mathbf{x}_i - \mathbf{M}(s_j, \theta_j)[\mathbf{x}_j] - \mathbf{t}_j) \tag{2}$$

where

$$M(s, \theta) \begin{bmatrix} x_{jk} \\ y_{jk} \end{bmatrix} = \begin{pmatrix} (s\cos\theta)x_{jk} - (s\sin\theta)y_{jk} \\ (s\sin\theta)x_{jk} + (s\cos\theta)y_{jk} \end{pmatrix}, \tag{3}$$

$$\mathbf{t} = (t_x, t_y, \ldots, t_x, t_y)^T \tag{4}$$

and $\mathbf{W}$ is a diagonal matrix of weights for each point. The weights in $\mathbf{W}$ are chosen such that points that remain stationary across the training set will be given larger weight in the sum. The weights for each point are determined using

$$w_k = (\sum_{l=0}^{n-1} V_{R_{kl}})^{-1} \tag{5}$$

where R_{kl} is the distance between two points k and l in a shape and $V_{R_{kl}}$ is the variance in this distance across the training images.

C. Statistical shape model

After the shapes are aligned via Procrustes analysis, PCA is used to create a statistical model that describes the variations in the training image shapes that cannot be described solely through rotations, translations, and scalings. Cootes et al. coined the term point distribution model (PDM) to describe the collection of landmark points and their subsequent variation in PC space after performing PCA. Fig. 2 illustrates the spread of landmark points from the 50 face images used to train the PDM. Note how certain landmarks, such as those around the nose, remain relatively stable across images, whereas the points at the chin or eyebrows are more prone to variation.

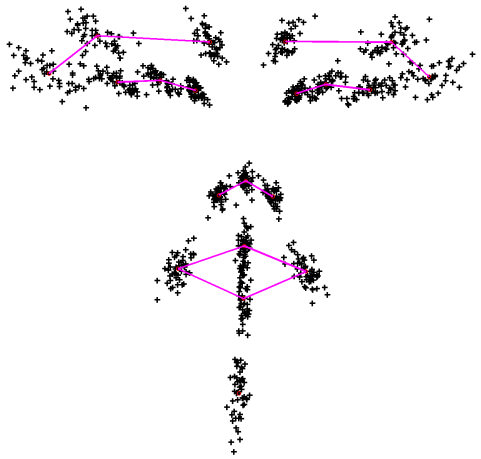

Fig. 2. Scatter plot of the landmark points from the 50 manually-labeled training images, with the mean shape overlaid.

As PCA is a standard statistical technique described in greater detail described at length in numerous sources, it will be described here only in brief, with focus on its relevance to the shape model. Each shape in the training set, after alignment, can be represented as a single point in $2n$ dimensional space. When plotted together in this hyperdimensional space, the shapes form a cloud of points assumed to be approximately ellipsoidal (i.e. Gaussian). PCA calculates the center of this hyperellipsoid (the mean shape) and its major axes. Variations within this ellipsoidal space represent realistic variations seen within the training data; when the model is used to generate new shapes, they are constrained to fall within this "Allowable Shape Domain," as Cootes et al. refer to it.

Proceeding with the description of PCA, the mean shape, $\bar{\mathbf{x}}$, of all N shapes is calculated using

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i. \tag{6}$$

The $2n \times 2n$ covariance matrix, $\mathbf{S}$, can be calculated as

$$\mathbf{S} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T. \tag{7}$$

The major axes of the hyperellipsoid are described by the eigenvectors, $\mathbf{p}_k$, of this covariance matrix (i.e. principal components), such that

$$\mathbf{S}\mathbf{p}_k = \lambda_k \mathbf{p}_k \tag{8}$$

where λ_k is the kth eigenvalue of S, and k ranges from 1 to $2n$.

New shapes can then be approximated as a linear combination of the mean shape and weighted principal components

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b} \tag{9}$$

where b is a vector of weights

$$\mathbf{b} = (b_1, b_2, \ldots, b_t)^T \tag{10}$$

with each weight constrained to a range determined by the particular PC's eigenvalue:

$$-3\sqrt{\lambda_k} \leq b_k \leq 3\sqrt{\lambda_k}. \tag{11}$$

D. Gray-level model

In addition to the shape model, it is necessary to train a model on the gray-level information in the pixels surrounding the landmark points in each training image. The gray-level model is used during the ASM search process to suggest new landmark positions as the shape is iteratively deformed to fit an object in the image. Following the work in [2], 2D gradient square profiles were used to build the gray-level, as opposed to the 1D derivative profiles suggested by Cootes et al. Square profiles were chosen here to avoid having to calculate normals to the boundary, which is made difficult by this paper's open contour arrangement of landmarks.

The process of determining the gradient profile at each landmark point can be described as follows:

1) Sample a 10×10 square region of the pixel values centered at the current landmark.
2) Compute the intensity gradient for the square region by convolution with the kernel

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

.
3) Normalize the matrix by dividing each element by the sum of the absolute values of each element.
4) Apply sigmoid equalization

$$x^{'} = \frac{x}{abs(x) + c} \tag{12}$$

where $x^{'}$ is a new element value and c determines the shape of the sigmoid. Sigmoid equalization compresses the values in a matrix, curtailing the effects of extreme lows or highs.

Fig. 3 shows an example of the gradient profiles for each landmark in an image downsampled by a factor of six.

After calculating the gradient profiles for each landmark in each image, the same PCA process that was used to generate the shape model is used to model the variations in gray levels across images. To accommodate the equations above, the gradient profile matrices are reshaped as column vectors, $\mathbf{g_1}, \ldots, \mathbf{g_n}$. As was described above for the shape model, the mean profile $\bar{\mathbf{g}}$ and the covariance matrix $\mathbf{S_g}$ are calculated from the gradient profiles, with the eigenvectors are denoted as $\mathbf{P_g}$.

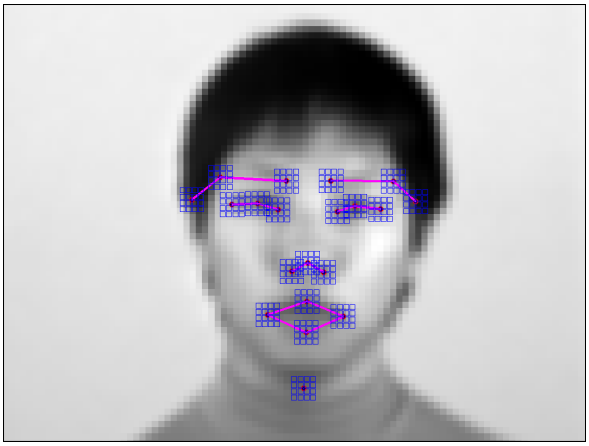

Fig. 3. 2D gradient profiles surrounding each landmark. The image is downsampled by a factor of six.

E. Multi-resolution

Implementing a multi-resolution approach to ASMs serves two purposes: improving the capture region of gross shape features and decreasing computation time. By starting the ASM algorithm at a low-resolution version of the original image and moving along the image pyramid to finer resolutions, the model is able to initially search for broad features (such as face location) before making fine adjustments to the face. The shape landmarks are assigned at full resolution and do not have a multi-resolution component. The gray profiles, however, are recalculated for each resolution level. The gray profiles for this implementation were calculated at six resolution levels, with downsampling factors of 1, 2, 3, 4, 5, and 6.

An example of the gradient profiles created for the central nose landmark at multiple resolutions is shown in Fig. 4. The ratios in the bottom right corner indicate the downsampling factor, with $1:6$ being the lowest resolution image.

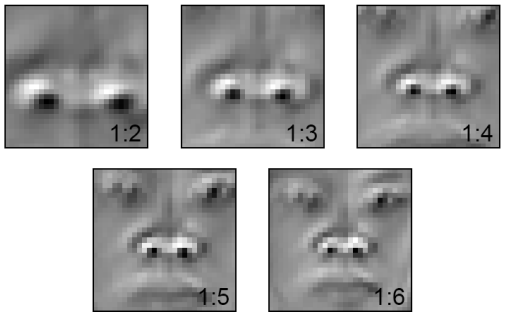

Fig. 4. 2D gradient profiles for the middle nose landmark, created at multiple resolutions. The ratio in the bottom corner denotes the downsampling factor.

F. Image search using active shape models

Once the shape and gray-level models are calculated for the N training images, the models can be used to search for objects

in new images. The details here will be specific to faces, but of course apply generally to whichever objects were contained within the training images.

The search process begins with a rough estimate of face location, typically performed by a global face estimation algorithm such as Viola-Jones[2]. Once the general position of the face is determined, the mean shape is placed centrally over the face and the iterative search process begins. As a general outline, the search process consists of estimating new positions for each landmark based on the gradient profile values, adjusting pose parameters (rotation, translation, and scaling) to match the mean shape to the suggested positions, and constraining the suggested points to fit within the constrains of the model, repeating the process until convergence. The specific details of the iterative ASM search are as follows:

1) For each landmark point, calculate nine 10×10 gray-level profiles. Each of the profiles is shifted to be centered at one of the nine pixels surrounding the landmark point in a 3×3 square. The optimal shift is to a pixel that produces a gradient profile closest to the mean profile, $\bar{\mathbf{g}}_\mathbf{i}$ for the current landmark, i.
2) Determine the weights, $\mathbf{b_g}$, that describe each of the new profiles, $\mathbf{g_{new}}$, in the gray-level PCA space
$$\mathbf{b_g} = {\mathbf{P_g}}^T(\mathbf{g_{new}} - \bar{\mathbf{g}}_\mathbf{i}). \tag{13}$$
3) Calculate F for the profile $\mathbf{g_{new}}$ at each shift, with F defined as
$$\sum_{j=1}^{t} \frac{b_{gj}^2}{\lambda_j} + \frac{2R^2}{\lambda_t} \tag{14}$$
where t is the number of PCs retained in the model and $R^2 = (\mathbf{g_{new}} - \bar{\mathbf{g}}_\mathbf{i})^T(\mathbf{g_{new}} - \bar{\mathbf{g}}_\mathbf{i}) - {\mathbf{b_g}}^T\mathbf{b_g}$, the sum of the squares of the difference between the model and the actual profile.
4) The pixel shift (within the 3×3 square) that produces a profile which minimizes F will be the suggested new location for the current landmark.
5) Using the Procrustes method described above, adjust the current shape's pose parameters to fit the new suggested positions.
6) Impose the shape model constraints on the posed positions. This step can be thought of as projecting the suggested positions into the model space, ensuring a valid shape. Equation (??) gives
$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Pb},$$
which can be used to generate new shapes. Equation 9 can be rearranged
$$\mathbf{b} = \mathbf{P}^T(\mathbf{x_{new}} - \bar{\mathbf{x}}) \tag{15}$$
and used to determine the PC weights, $\mathbf{b}$ for the new suggested shape. The weights are constrained by equation 11 to ensure a valid shape.
7) Perform Procrustes analysis once more, to transfer the new shape from the model space to the image space.

The steps above are performed at each resolution level, from course to fine. The steps are repeated until convergence, typically within four iterations.

G. Evaluation

In order to determine the quality of the face detection implementation, the models were retrained on only 40 of the 50 manually-labeled images. The remaining 10 images were used to compare between the shape found by the model and the manually-labeled ground truth. A qualitative assessment was performed on approximately 30 images that were not a part of the training set. The mean shape was initialized using point and click at various locations away from the face, and performance was monitored in terms of detection speed and accuracy.

III. Results

The ASM algorithm implemented here was able to successfully detect and conform to the features of nearly every face within a 90 image test set. Fig. 5 demonstrates the results of an iterative search process for an example image.

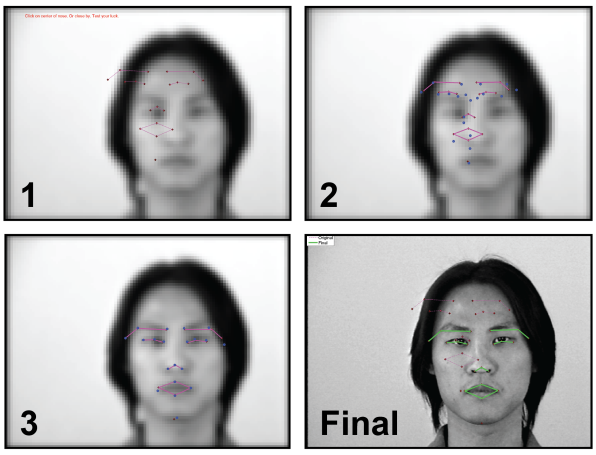

Fig. 5. ASM search at four different iterations of the search process for an example image. The bottom right image is displayed at full resolution with the final shape convergence.

Fig. 6 displays the correlations between the position found by the search algorithm and the manually-labeled ground truth for six landmark points from 10 test images. The landmark points are numbers 2, 5, 7, 12, 14, and 20 from 1, corresponding to the left iris, right iris, left eyebrow edge, right eyebrow edge, middle of nose, and bottom of chin, respectively. A strong correlation is apparent when comparing between the red least squares regression line and the black $y = x$ line in each scatter plot.

A. Retinal layer segmentation

Once the ASM implementation was shown to be functional with face detection, it was trained on a set of ocular coherence tomography (OCT) scans of the human retina. The results of the attempted segmentation are shown in 7. The ASM model did not perform well in any of the OCT test cases, as evidenced by the erroneous segmentations shown in the figure.

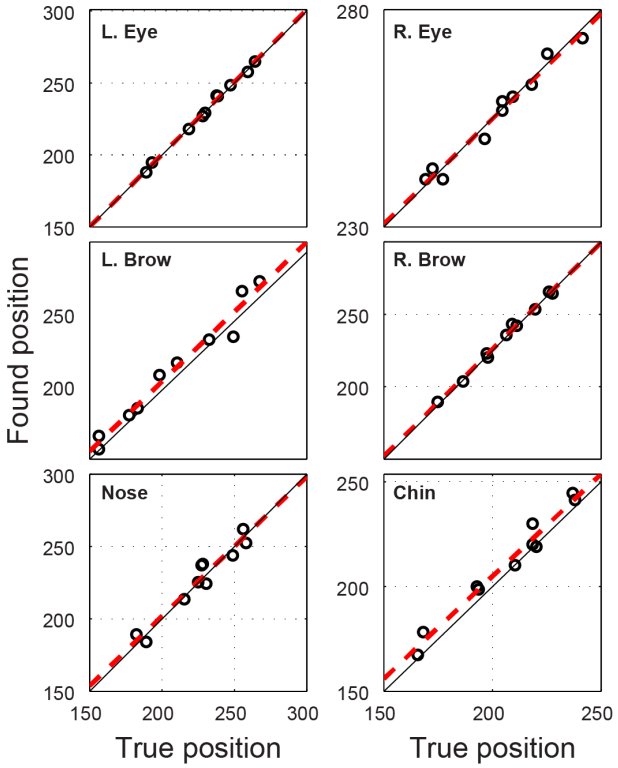

Fig. 6. Scatter plots comparing the found and true locations for six different landmark points, from 10 test images. The red line in each scatter plot is the least squares regression line for those data.

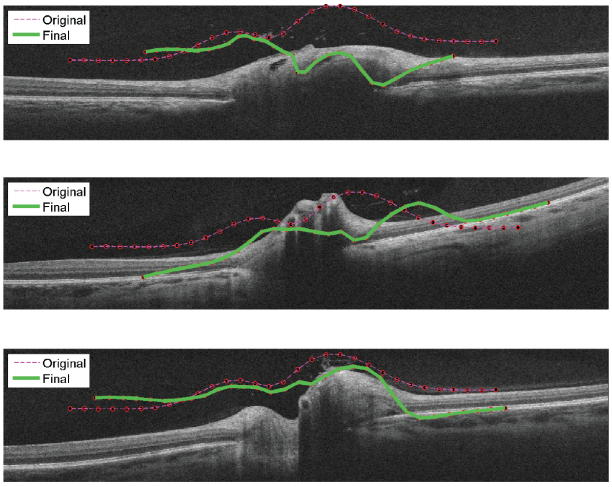

Fig. 7. Attempted automatic segmentation of retinal layers using ASMs.

IV. Discussion

A. Shortcomings and improvements

While ASMs are able to do quite a good job of object recognition given their relatively simple implementation, the technique is now rather outdated and its shortcomings have been addressed in a number of ways.

One major shortcoming of the original ASM approach is its assumption that the different shapes within the training set are distributed normally and can thus be described with PCA, which is a Gaussian model. Many object sets, especially within medical imaging, will not follow a Gaussian distribution and therefore may not be characterized well by ASMs. The gray-level models will be similarly effected. An example of a nonlinear model being used to overcome this limitation of ASMs can be found in [3].

The ASM technique also depends on a rather accurate estimation of global face position before the algorithm can search for specific features. Some model parameters, such as the 2D gray-profile search size can be manipulated to improve detection of global features, but the technique largely relies on the accuracy of separate global object estimation algorithms.

The ASM technique also relies on manual labeling for the training set. This is a time-consuming process that would ideally be automated. One such approach would be to combine the ASM algorithm with a segmentation approach such as graph cut, that could automate the placement of landmarks within training images.

Finally, the ASM technique is generally slow by today's standards. One possible area of improvement would be to trim sparse matrices in the gray-profiles as described by [2].

B. Future directions

The next step for this project will be to run the ASM implementation on other data sets. Apart from the visualization functions, the Matlab code in this implementation is completely independent of faces. It is therefore a trivial process to extend the implementation to other object search tasks to compare performance. Simply exposing the shape and gray models to a larger set of faces in different lightings and poses would likely improve the capture capabilities of the current implementation. Another important next step in the implementation of this ASM algorithm is to improve the quantitative assessment of accuracy. It is clear from a qualitative perspective that the algorithm is able to detect local features in faces, but a more rigorous approach is required before the implementation shown here can be compared to other algorithms. Training sets with additional landmark arrangements may also prove effective when localizing facial features.

While it was not detailed at length in this paper, there were some efforts made to use this ASM implementation to detect retinal layers in OCT scans. The segmentation performance was subpar, largely due to the variance and heterogeneity of the training set. It may be worth looking into the potential of ASMs as a shape analysis tool for retinal layers, but its segmentation utility in this setting appears to be limited.

V. Conclusion

The work shown here is an implementation of the original ASM method by Cootes et al. [1] with modifications to the gray-profiles from [2]. On both a qualitative and quantitative level, the implementation was shown to be able to detect, with high accuracy, local facial features in grayscale images.

The implementation here sets the foundation for future work incorporating nonlinear models, additional training sets, and varied landmark arrangements. The ASM technique, while outdated, was still shown to be effective at identifying facial features.

ACKNOWLEDGMENT

The author would like to thank Professor Jacob for his help interpreting the original implementation of the ASM algorithm and his engaging lectures throughout the semester.

REFERENCES

[1] 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.

[2] Stephen Milborrow. *Multiview Active Shape Models with SIFT Descriptors*. PhD thesis, University of Cape Town, 2016.

[3] Bram Van Ginneken, Alejandro F. Frangi, Joes J. Staal, Bart M. Ter Haar Romeny, and Max A. Viergever. Active shape model segmentation with optimal features. *IEEE Transactions on Medical Imaging*, 21(8):924–933, 2002.

John W. Miller is a first year masters student at the University of Iowa, studying Electrical and Computer Engineering.