Locating Facial Features with an Extended Active Shape Model

Stephen Milborrow and Fred Nicolls

Department of Electrical Engineering
University of Cape Town, South Africa
www.milbo.users.sonic.net

Abstract. We make some simple extensions to the Active Shape Model of Cootes et al. [4], and use it to locate features in frontal views of upright faces. We show on independent test data that with the extensions the Active Shape Model compares favorably with more sophisticated methods. The extensions are (i) fitting more landmarks than are actually needed (ii) selectively using two- instead of one-dimensional landmark templates (iii) adding noise to the training set (iv) relaxing the shape model where advantageous (v) trimming covariance matrices by setting most entries to zero, and (vi) stacking two Active Shape Models in series.

1 Introduction

Automatic and accurate location of facial features is difficult. The variety of human faces, expressions, facial hair, glasses, poses, and lighting contribute to the complexity of the problem.

This paper focuses on the specific application of locating features in unobstructed frontal views of upright faces. We make some extensions to the Active Shape Model (ASM) of Cootes et al. [4] and show that it can perform well in this application.

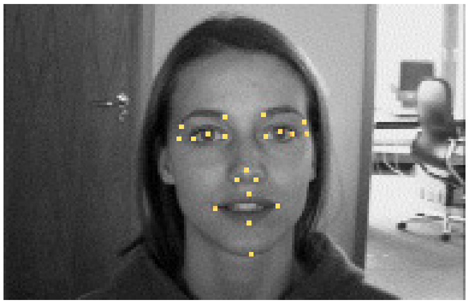

Fig. 1. A face with correctly positioned landmarks. This image is from the BioID set [15].

2 Active Shape Models

This section describes Active Shape Models [8].

A *landmark* represents a distinguishable point present in most of the images under consideration, for example, the location of the left eye pupil (Fig. 1). We locate facial features by locating landmarks.

A set of landmarks forms a shape. Shapes are represented as vectors: all the x- followed by all the y-coordinates of the points in the shape. We align one shape to another with a similarity transform (allowing translation, scaling, and rotation) that minimizes the average euclidean distance between shape points. The *mean shape* is the mean of the aligned training shapes (which in our case are manually landmarked faces).

The ASM starts the search for landmarks from the mean shape aligned to the position and size of the face determined by a global face detector. It then repeats the following two steps until convergence (i) suggest a tentative shape by adjusting the locations of shape points by template matching of the image texture around each point (ii) conform the tentative shape to a global shape model. The individual template matches are unreliable and the shape model pools the results of the weak template matchers to form a stronger overall classifier. The entire search is repeated at each level in an image pyramid, from coarse to fine resolution.

It follows that two types of submodel make up the ASM: the *profile model* and the *shape model*.

The profile models (one for each landmark at each pyramid level) are used to locate the approximate position of each landmark by template matching. Any template matcher can be used, but the classical ASM forms a fixed-length normalized gradient vector (called the *profile*) by sampling the image along a line (called the *whisker*) orthogonal to the shape boundary at the landmark. During training on manually landmarked faces, at each landmark we calculate the mean profile vector $\bar{\mathbf{g}}$ and the profile covariance matrix $\mathbf{S_g}$. During searching, we displace the landmark along the whisker to the pixel whose profile $\mathbf{g}$ has lowest Mahalanobis distance from the mean profile $\bar{\mathbf{g}}$:

$$MahalanobisDistance = (\mathbf{g} - \bar{\mathbf{g}})^{\mathbf{T}} \mathbf{S_g^{-1}} (\mathbf{g} - \bar{\mathbf{g}}). \tag{1}$$

The shape model specifies allowable constellations of landmarks. It generates a shape $\hat{\mathbf{x}}$ with

$$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{\Phi b} \tag{2}$$

where $\bar{\mathbf{x}}$ is the mean shape, $\mathbf{b}$ is a parameter vector, and $\mathbf{\Phi}$ is a matrix of selected eigenvectors of the covariance matrix $\mathbf{S_s}$ of the points of the aligned training shapes. Using a standard principal components approach, we model as much variation in the training set as we want by ordering the eigenvalues λ_i of $\mathbf{S}_s$ and keeping an appropriate number of the corresponding eigenvectors in $\mathbf{\Phi}$. We use a single shape model for the entire ASM but scale it for each pyramid level.

We can generate various shapes with Equation 2 by varying the vector parameter $\mathbf{b}$. By keeping the elements of $\mathbf{b}$ within limits (determined during model building) we ensure that generated face shapes are lifelike.

Conversely, given a suggested shape $\mathbf{x}$, we can calculate the parameter $\mathbf{b}$ that allows Equation 2 to best approximate $\mathbf{x}$ with a model shape $\hat{\mathbf{x}}$. Cootes and Taylor [8] describe an iterative algorithm that gives the $\mathbf{b}$ and $\mathbf{T}$ that minimizes

$$distance(\mathbf{x},\ \mathbf{T}(\bar{\mathbf{x}} + \mathbf{\Phi}\mathbf{b})) \tag{3}$$

where $\mathbf{T}$ is a similarity transform that maps the model space into the image space.

3 Related Work

Active Shape Models belong to the class of models which after a shape is situated near an image feature interact with the image to warp the shape to the feature. They are deformable models likes snakes [16], but unlike snakes they use an explicit shape model to place global constraints on the generated shape. ASMs were first presented by Cootes et al. [3]. Cootes and his colleagues followed with a succession of papers cumulating in the classical ASM described above [8] [4].

Many modifications to the classical ASM have been proposed. We mention just a few. Cootes and Taylor [6] employ a shape model which is a mixture of multivariate gaussians, rather than assuming that the shapes come from the single gaussian distribution implicit in the shape model of the classical ASM. Romdhani et al. [22] use Kernel Principal Components Analysis [23] and a Support Vector Machine. Their software trains on 2D images, but models non-linear changes to face shapes as they are rotated in 3D. Rogers and Graham [21] robustify ASMs by applying robust least-squares techniques to minimize the residuals between the model shape and the suggested shape. Van Ginneken et al. [12] take the tack of replacing the 1D normalized first derivative profiles of the classical ASM with local texture descriptors calculated from “locally orderless images” [17]. Their method automatically selects the optimum set of descriptors. They also replace the classical ASM profile model search (using Mahalanobis distances) with a k-nearest-neighbors classifier. Zhou et al. [25] estimate shape and pose parameters using Bayesian inference after projecting the shapes into a tangent space. Li and Ito [24] build texture models with AdaBoosted histogram classifiers. The Active Appearance Model [5] merges the shape and profile model of the ASM into a single model of appearance, and itself has many descendants. Cootes et al. [7] report that landmark localization accuracy is better on the whole for ASMs than AAMs, although this may have changed with subsequent developments to the AAM.

4 Extensions to the ASM

We now look at some extensions to the classical ASM. Figure 3 (Sec. 5.1) shows the increase in performance for each of these extensions.

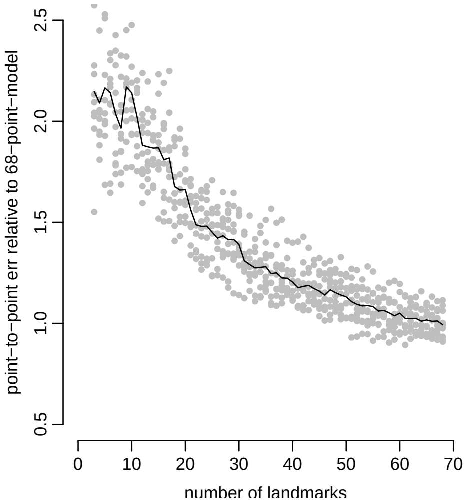

Fig. 2. Mean error versus number of landmarks.

4.1 Number of Landmarks

A straightforward way to improve the mean fit is to increase the number of landmarks in the model (Fig. 2). Fitting a landmark tends to help fitting other landmarks, so results are improved by fitting more landmarks than are actually needed. Search time increases roughly linearly with the number of landmarks.

Fig. 2 was constructed as follows from the XM2VTS [19] set of manually landmarked faces. For a given number (from 3 to 68) of landmarks, that number of landmarks was chosen randomly from the 68 in the XM2VTS test. With the chosen landmarks, a model was built and tested to give one gray dot. This was repeated ten times for each number of landmarks. The black line shows the mean error for each number of landmarks.

4.2 Two Dimensional Profiles

The classical ASM uses a one-dimensional profile at each landmark, but using two-dimensional "profiles" can give improved fits. Instead of sampling a one-dimensional line of pixels along the whisker, we sample a square region around the landmark. Intuitively, a 2D profile area captures more information around the landmark and this information if used wisely should give better results.

During search we displace the sampling region in both the "x" and "y" directions, where x is orthogonal to the shape edge at the landmark and y is tangent to the shape edge. We must rely on the face being approximately upright because 2D profiles are aligned to the edges of the image. The profile covariance matrix $\mathbf{S_g}$ of a set of 2D profiles is formed by treating each 2D profile matrix as a long vector (by appending the rows end to end), and calculating the covariance of the vectors.

Any two dimensional template matching scheme can be used, but the authors found that good results were obtained using gradients over a 13x13 square around the landmark, after prescaling faces to a constant width of 180 pixels. The values 13 and 180 were determined during model building by measurements on a validation set, as were all parameter values in this paper (Sec. 5).

Gradients were calculated with a 3x3 convolution mask ((0,0,0),(0,-2,1),(0,1,0)) and normalized by dividing by the Frobenius norm of the gradient matrix. The effect of outliers was reduced by applying a mild sigmoid transform to the elements x_i of the gradient matrix: $x'_i = x_i/(abs(x_i) + constant)$.

Good results were obtained using 2D profiles for the nose and eyes and surrounding landmarks, with 1D profiles elsewhere.

4.3 Adding Noise During Training

The XM2VTS set used for training (Sec. 5) contains frontal images of mostly caucasian working adults and is thus a rather limited representation of the variety of human faces. A shape model built with noise added to the training shapes helps the trained model generalize to a wider variety of faces. Good results can be obtained with the following techniques:

1. Add gaussian noise with a standard deviation of 0.75 pixels to the x- and y-positions of each training shape landmark. In effect, this increases variability in the training set face shapes.
2. Randomly choose the left or the right side each face. Generate a stretching factor ϵ for each face from a gaussian distribution with a standard deviation of 0.08. Stretch or contract the chosen side of the face by multiplying the x position (relative to the face center) of each landmark on that side by $1 + \epsilon$. This is roughly equivalent to rotating the face slightly.

4.4 Loosening Up the Shape Model

In Equation 2, the constraints on the generated face shape are determined by the number of eigenvectors n_{eigs} in $\mathbf{\Phi}$ and the maximum allowed values of elements in the parameter vector $\mathbf{b}$. When conforming the shape suggested by the profile models to the shape model, we clip each element b_i of $\mathbf{b}$ to $b_{max}\sqrt{\lambda_i}$ where λ_i is the corresponding eigenvalue The parameters n_{eigs} and b_{max} are global constants determined during model building by parameter selection on a validation set. See [8] for details.

The profile models are most unreliable when starting the search (for example, a jaw landmark can snag on the collar), but become more reliable as the search progresses. We can take advantage of this increase in reliability with two modifications to the standard ASM procedure described above. The first modification sets n_{eigs} and b_{max} for the final pyramid level (at the original image scale) to larger values. The second sets n_{eigs} and b_{max} for the final iteration at each pyramid level to larger values. In both cases the landmarks at that stage of the search tend to be already positioned fairly accurately, for the given pyramid level. It is therefore less likely that the profile match at any landmark is grossly mispositioned, allowing the shape constraints to be weakened.

These modifications are effective for 2D but not for 1D profiles. The 1D profile matches are not reliable enough to allow the shape constraints to be weakened.

4.5 Trimming the Profile Covariance Matrices

For 2D profiles, calculation of the Mahalanobis distances dominates the overall search time. We can reduce this time (with little or no effect on landmark location accuracy) by "trimming" the covariance matrix

The covariance between two pixels in a profile tends to be much higher for pixels that are closer together. This means that we can ignore covariances for pixels that are more than 3 pixels apart, or equivalently clear them to 0. Clearing elements of a covariance matrix may result in a matrix that is no longer positive definite (which is necessary for a meaningful Mahalanobis distance calculation in Equation 1). We therefore adjust the trimmed matrix to a "nearby" positive definite matrix. This can be done by iterating the following procedure a few times: perform a spectral decomposition of the trimmed covariance matrix $A = Q\Lambda Q^T$, set zero or negative eigenvalues in Λ to a small positive number, reconstruct the matrix from the modified Λ, and re-trim. A suitable "small positive number" is $iter\_nbr \times abs(min(eig\_vals(A)))$. More rigorous ways of forcing positive definiteness are presented in Gentle [11] and in Bates and Maechler [1].

Trimming the covariance matrices in conjunction with a sparse matrix multiplication routine roughly halves the overall search time.

4.6 Stacking Models

Accurate positioning of the start shape is crucial — it is unlikely that an ASM search will recover completely from a bad start shape. One way of better positioning the start shape is to run two ASM searches in series, using the results of the first search as the start shape for the second search. In practice is suffices to use 1D profiles for the first model and to start the second model at pyramid level 1, one level below full size. Stacking helps the worst fits, where the start shape is often badly mis-positioned, but has little effect where the start shape is already well positioned.

5 Experimental Results

Before giving experimental results we briefly review model assessment in more general terms [13]. The overall strategy for selecting parameters is

```
1. for each model parameter
2.     for each parameter value
3.         train on a set of faces
4.         evaluate the model by using it to locate landmarks
5.     select the value of the parameter that gives the best model
6. test the final model by using it to locate landmarks.
```

Two processes are going on here: model selection which estimates the performance of different models in order to choose one (steps 2-5 above), and model assessment which estimates the final model's performance on new data (step 6 above). We want to measure the generalization ability of the model, not its ability on the set it was trained on, and therefore need three independent datasets (i) a *training set* for step 3 above (ii) a parameter selection or *validation set* for step 4 above, and (iii) a *test set* for step 6 above.

For the training set we used the XM2VTS [19] set. We effectively doubled the size of the training set by mirroring images, but excluded faces that were of poor quality (eyes closed, blurred, etc.).

For the validation set we used the AR [18] set. So, for example, we used the AR set for choosing the amount of noise discussed in section 4.3. We minimized overfitting to the validation set by using a different subset of the AR data for selecting each parameter. Subsets consisted of 200 randomly chosen images.

For the test set we used the BioID set [15]. More precisely, the test set is those faces in the BioID set that were successfully found by the OpenCV [14] implementation of the Viola-Jones face detector (1455 faces, which is 95.7% of the total 1521 BioID faces).

We used manual landmarks for these three sets from the FGNET project [9].

Cross validation on a single data set is another popular approach. We did not use cross validation because three datasets were available and because of the many instances of near duplication of images within each dataset.

Following Cristinacce [10], we present results in terms of the *me17* measure. The me17 is calculated by taking the mean of the euclidean distances between each of the 17 internal face points located by the search and the corresponding manually landmarked point. This mean is normalized by dividing by the distance between the manually landmarked eye pupils. We use only 17 of the 20 manually landmarked BioID points because the 3 points near the sides of the face have a high variability across human landmarkers.

5.1 Relative Performance

Fig. 3 summarizes and compares results from applying each of the modifications described in this paper. Each graph point represents the me17 averaged over all faces in the test set, for the given model. Each model incorporates the improvements of the models to its left but not to its right.

For example, the entry labeled **4.2 2D profiles** shows results for the model described in section 4.2. The model uses the 2D profiles described in that section and incorporates the techniques prior to but not subsequent to section 4.2 The

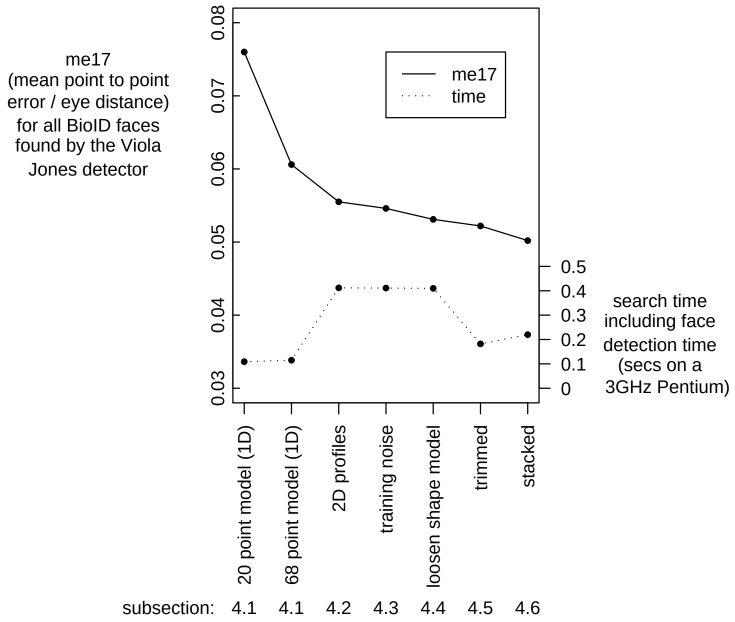

Fig. 3. Relative performance of various models.

graph shows that using 2D profiles decreases the me17 from 0.061 to 0.055 but increases the search time from 110 to 410 ms.

The mean me17 of the final stacked model is 66% of the initial 20 point model. The biggest single improvement comes from adding more points to the model, followed by using 2D profiles, followed by stacking. A different test set or different evaluation order would give somewhat different results, but the graph is representative of the relative performance of the various modifications.

5.2 Comparison to Previously Published Results

Fig. 4 compares the best model in this paper, the stacked model (section 4.6), to the Constrained Local Model presented in Cristinacce and Cootes [10]. Briefly, the Constrained Local Model is similar to an Active Appearance Model [5], but instead of modeling texture across the whole face it models a set of local feature templates. During search, the feature templates are matched to the image using an efficient shape constrained search. The model is more accurate and more robust than the original Active Appearance Model.

The results in Cristinacce and Cootes' paper appear to be the best previously published facial landmark location results and are presented in terms of the me17 on the BioId set, which makes a direct comparison possible. The dotted curve in Fig. 4 reproduces the curve in Fig. 4(c) in their paper. The figure shows that the stacked model on independent data outperforms the Constrained Local Model.

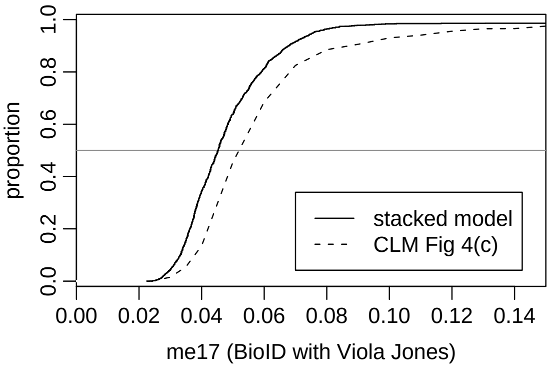

Fig. 4. Comparison to Constrained Local Model [10]

The median me17 for the stacked model is 0.045 (2.4 pixels), the best me17 is 0.0235 (1.4 pixels), and the worst is 0.283 (14 pixels). The long right hand tail of the error distribution is typical of ASMs.

6 Conclusion and Future Work

This paper presented some modifications to the Active Shape Model which make it competitive with more sophisticated methods of locating features in frontal views of upright faces.

A few simple rules of thumb for improving ASMs became apparent. You can get better fits by adding more landmarks. You can discard most elements of the covariance matrices for increased speed without loss of quality. You get better results with a better start shape, and you can do this by running two models in series.

The techniques used in this paper are fairly standard. Perhaps the main contribution of the paper is assembling them together in a sound fashion. Advantages of the techniques are their simplicity and applicability for use in conjunction with other methods. For example, extra landmarks and stacked models would possibly improve the performance of the Constrained Local Model shown in Fig. 4.

The results are still not as good as manual landmarks. Further work will investigate combining multiple profiling techniques at each landmark with a decision tree [2] or related method. Here the training process would try different profiling techniques at each landmark and build a decision tree (for each landmark) that would select or combine techniques during searching.

Additional documentation and source code to reproduce the results in this paper can be found at this project's web site [20].

References

1. D. Bates, M. Maechler: Matrix: A Matrix package for R (2008). cran.r-project.org/web/packages/Matrix/index.html. See the nearPD function in this R package for methods of forcing positive definiteness.
2. Breiman, Friedman, Olshen, Stone: Classification and Regression Trees. Wadsworth (1984)
3. T. F. Cootes, D. H. Cooper, C. J. Taylor, J. Graham: A Trainable Method of Parametric Shape Description. BMVC 2, 54–61 (1991)
4. T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham: Active Shape Models — their Training and Application. CVIU 61, 38–59 (1995)
5. T. F. Cootes, G. J. Edwards, C. J. Taylor: Active Appearance Models. ECCV 2, 484–498 (1998)
6. T. F. Cootes, C. J. Taylor: A Mixture Model for Representing Shape Variation. Image and Vision Computing 17 8, 567–574 (1999)
7. T. F. Cootes, G. J. Edwards, C. J. Taylor: Comparing Active Shape Models with Active Appearance Models. Proc. British Machine Vision Conference (ed T.Pridmore, D.Elliman), 1, 173–182 (1999)
8. T. F. Cootes, C. J. Taylor: Technical Report: Statistical Models of Appearance for Computer Vision. The University of Manchester School of Medicine. (2004) www.isbe.man.ac.uk/ bim/refs.html
9. T. F. Cootes et al.: FGNET manual annotation of face datasets. (2002) www-prima.inrialpes.fr/FGnet/html/benchmarks.html
10. D. Cristinacce, T. Cootes: Feature Detection and Tracking with Constrained Local Models. BMVC 17, 929–938 (2006)
11. J. E. Gentle: Numerical Linear Algebra for Applications in Statistics. Springer (1998). See page 178 for methods of forcing positive definiteness.
12. B. van Ginneken, A. F. Frangi, J. J. Stall, B. ter Haar Romeny: Active Shape Model Segmentation with Optimal Features. IEEE-TMI 21, 924–933 (2002)
13. T. Hastie, R. Tibshirani, J. Friedman: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer (2003). See chapter 7 for methods of model assessment.
14. Intel: Open Source Computer Vision Library. Intel (2007)
15. O. Jesorsky, K. Kirchberg, R. Frischholz: Robust Face Detection using the Hausdorff Distance. AVBPA 90–95 (2001)
16. M. Kass, A. Witkin, D. Terzopoulos: Snakes: Active Contour Models. IJCV 1, 321–331 (1987)
17. J. J. Koenderink, A. J. van Doorn: The Structure of Locally Orderless Images. IJCV 31 2/3, 159–168 (1999)
18. A. M. Martinez, R. Benavente. The AR Face Database: CVC Tech. Report 24, (1998)
19. K. Messer, J. Matas, J. Kittler, J. Luettin, G. Maitre: XM2VTS: The Extended M2VTS Database. AVBPA (1999)
20. S. Milborrow: Stasm software library (2007). www.milbo.users.sonic.net/stasm
21. M. Rogers, J. Graham: Robust Active Shape Model Search. ECCV 4, 517–530 (2002)
22. S. Romdhani, S. Gong, A. Psarrou: A Multi-view Non-linear Active Shape Model using Kernel PCA. BMVC 10, 483–492 (1999)
23. S. Scholkopf, A. Smola, K. Muller: Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation 10 5, 1299-1319 (1998)

24. Yuanzhong Li, Wataru Ito: Shape Parameter Optimization for AdaBoosted Active Shape Model. ICCV 1, 251–258 (2005)
25. Y. Zhou, L. Gu, H. J. Zhang: Bayesian Tangent Shape Model: Estimating Shape and Pose Parameters via Bayesian Inference. CVPR (2003)