

# Boosted Regression Active Shape Models

David Cristinacce and Tim Cootes  
Dept. Imaging Science and Biomedical Engineering  
University of Manchester, Manchester, M13 9PT, U.K.  
david.cristinacce@manchester.ac.uk

## Abstract

We present an efficient method of fitting a set of local feature models to an image within the popular Active Shape Model (ASM) framework [3]. We compare two different types of non-linear boosted feature models trained using GentleBoost [9]. The first type is a conventional feature detector classifier, which learns a discrimination function between the appearance of a feature and the local neighbourhood. The second local model type is a boosted regression predictor which learns the relationship between the local neighbourhood appearance and the displacement from the true feature location. At run-time the second regression model is much more efficient as only the current feature patch needs to be processed. We show that within the local iterative search of the ASM the local feature regression provides improved localisation on two publicly available human face test sets as well as increasing the search speed by a factor of eight.

## 1 Introduction

We describe a method of fitting a model of an object class to new images containing unseen examples. In this paper the class of objects is the human face, however the method can be applied to any type of object with corresponding features between different examples, for instance most types of medical images and many man-made objects.

This model based approach to computer vision requires a labelled set of training examples, with corresponding features between images (see Figure 1 for examples from our human face training set). There are many different types of models, most of which encode the appearance variation around or within the labelled region and also encode the shape variation of the feature locations across the training set [2, 3, 4, 5, 7].

This paper uses the Active Shape Model (ASM) framework due to Cootes *et al.* [3]. The ASM models shape variation across the training set with a statistical shape model and an individual model for each local feature. At run-time each local model updates its estimate of the best local match and the shape model is fitted to the full set of point estimates to eliminate false positive matches.

The original ASM paper [3] used local eigen patches [15] to model each feature. However in this paper we use non-linear boosted features trained using GentleBoost [9]. We investigate local feature detection using boosted features and also boosted regression, which aims to predict local feature points without the need for a sliding window search in the local neighbourhood. The boosted regression approach is shown to out perform local

feature detection when applied to the publicly available BIOID [10] and XM2VTS [13] data sets. The boosted regression approach is extremely fast, able to perform local search at  $> 60$  frames per second and also able to achieve results comparable to other published methods [4].

## 2 Background

Active shape models are a method of modelling shape variation across a training set of labelled examples (see Cootes *et al.* [3]). The shape model can be fitted to a set of feature detections to remove outliers. There are various other shape constraint methods, such as the tree structure used in the Pictorial Structure Matching method due to Felzenszwalb and Huttenlocher [7] or the softer shape model constraint used by Cristinacce and Cootes [4] which take into account the local feature responses when fitting the shape model. However the ASM is a simple method which we use here to compare the performance of local regression versus detection models.

The choice of possible feature detection methods to use in the ASM is large. For example normalised correlation patches have been shown to be successful when combined with a generative model of appearance [4]. Other varieties of feature detectors are Local Binary Patterns [1], mutual information [5], Boosted Haar Wavelets [17] and K-Nearest Neighbour Classifiers [16]. The original ASM algorithm used local eigen models [15], but here we use discriminative haar wavelets trained using GentleBoost [9], as this technique has shown to work extremely well for whole face detection [12].

An alternative to feature detection methods for local search are regression techniques. For example the well known Active Appearance Model AAM algorithm [2] fits a deformable generative model to a patch of the image and then performs linear regression on the texture residual to update the internal model parameters and thus perform a local search. The AAM models the whole object, whereas our proposed method uses local features.

Another example of feature finding using a regression method is Zheng *et al.* [19], who use Rankboost [8] to rank the possible image warpings from the mean shape to the an unseen image and thus compute feature points. They present good results on manually cropped Echo Cardiograms and Face Photographs. Everingham *et al.* compare Kernel Ridge Regression with a Bayesian Classifier approach, but report better results with the simple classifier method for the task of eye finding [6].

A recent approach to using local regression models is described by Wimmer *et al.* [18], who train model trees to regress from local haar wavelet features to a objective function designed to peak at the true feature location. At run-time this allows the best matching location to be predicted for each feature. Langs *et al.* [11] use canonical correlation analysis to perform an AAM style search with filter responses located at individual feature points. Seise *et al.* [14] use the ASM framework in conjunction with a Relevance Vector Machine (RVM) regressor to update each feature location.

Our approach is similar to the approach of Wimmer and Seise, but uses GentleBoost as the regression function to predict the current displacement for each feature. We make a comparison between local regression methods and feature classifiers trained on the same data, both using the GentleBoost framework [9]. In Section 3 we describe our implementation in more detail and in Section 4 show that the regression method gives improved

localisation performance, compared to the boosted classifier, but at much lower computational cost.

### 3 Methodology

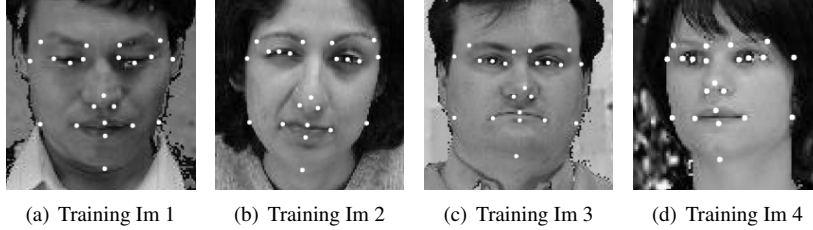


Figure 1: Manually Labelled Training Images

#### 3.1 Active Shape Model

The Active Shape Model (ASM) was introduced by Cootes *et al.* [3] as a method of fitting a set of local feature detectors to an object and simultaneously taking into account global shape considerations. The allowable shape deformations are learnt from a manually labelled training set (see Figure 1) to produce a linear shape model with the following form:-

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}_s \mathbf{b}_s \quad (1)$$

Where  $\bar{\mathbf{x}}$  is the mean shape,  $\mathbf{P}_s$  is a set of orthogonal modes of variation and  $\mathbf{b}_s$  is a set of shape parameters. Given a set of hypothesised feature points  $\mathbf{Y}$  in the image plane the shape model parameters  $\mathbf{b}_s$  can be determined by minimising

$$|\mathbf{Y} - T_t(\bar{\mathbf{x}} + \mathbf{P}_s \mathbf{b}_s)| \quad (2)$$

By placing constraints on the allowable shape parameters  $\mathbf{b}_s$  the shape model estimate of the current feature points  $T_t(\bar{\mathbf{x}} + \mathbf{P}_s \mathbf{b}_s)$  are constrained to form a plausible shape.

The shape model is active in the sense that feature detectors are applied to search in the local neighbourhood of each point and the best match of each detector is recorded. Assuming the majority of the detections are correct, the shape model can be fitted to this set of points and outlier detections discarded. This constraint on feature matching has been shown to improve results compared to merely taking the best unconstrained fit of each feature [3].

#### 3.2 Boosted Feature Detection

Any set of feature detectors can be used in the ASM framework described above. The original algorithm [3] used eigen model [15] profiles of the texture about each of the individual feature points. In this work we choose boosted feature detectors, which have

a similar formulation to the well known Viola and Jones face detector [17]. The training method we use is GentleBoost [9], which has shown to give superior performance compared to the original AdaBoost algorithm for the task of face detection [12].

---

**Algorithm 1** Gentle Boost Training Algorithm - Classification [9]

---

1. Start with weights  $w_i = 1/N$ ,  $i = 1, 2, \dots, N$ ,  $F(x) = 0$  and  $y_i = 1$  for positive examples,  $y_i = -1$  for negative examples.
  2. Repeat for  $m = 1, 2, \dots, M$ :
    - (a) Fit all the regression functions  $f_m(x)$  by weighted least squares of  $y_i$  to  $x_i$  with weights  $w_i$ .
    - (b) Select the  $f_m(x)$  with least weighted error  $\sum_{i=1}^N (w_i(y_i - f_m(x_i)))^2$
    - (c) Update  $F(x) \leftarrow F(x) + f_m(x)$
    - (d) Update  $w_i \leftarrow w_i \exp(-y_i f_m(x_i))$  and re-normalise
  3. Output the classifier  $\text{sign}[F(x)] = \text{sign}[\sum_{m=1}^M f_m(x)]$
- 

The GentleBoost classifier training procedure is described in Algorithm 1. The aim of the algorithm is to learn a discrimination function between a set of positive and negative examples. Where positive examples are image patches centred on the correct feature locations and negative examples are nearby examples displaced from the true locations, see Figure 2.

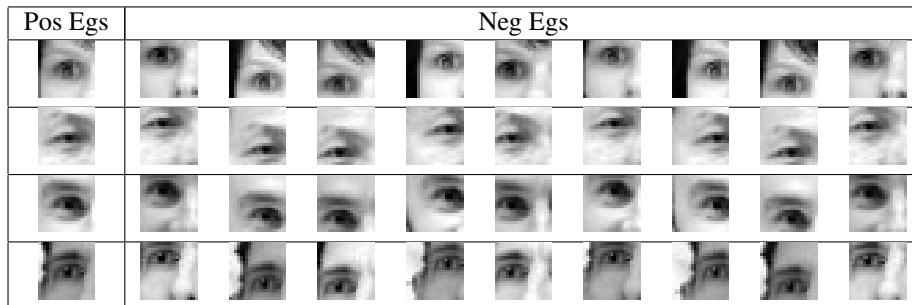


Figure 2: Positive and Negative Examples for right eye detector

Following the notation in Algorithm 1, each positive training patch  $x_i$  has label  $y_i = +1$  and each negative patch has label  $y_i = -1$ . To train using GentleBoost it is necessary to select a family of functions  $f(x)$  which take an image patch  $x_i$  and attempt to predict the classification  $y_i$  for a given set of training weights  $w_i$ . In this paper  $f(x)$  is a binned histogram of responses from a haar wavelet (we use the same set as [17]). Each  $f(x)$  is trained by computing the weighted mean of target values  $y_i$  in each histogram bin. The error for each  $f(x)$  is the weighted sum of square differences between the target value  $y_i$  and the mean of the selected bin determined by the wavelet response to patch  $x_i$ . The GentleBoost training algorithm selects a set of weak classifier functions and outputs a

strong classifier, as described in Algorithm 1. The training algorithm is computationally expensive, as the weak classifier functions  $f(x)$  have to be retrained at every iteration with new weights  $w_i$ .

There are also several parameters that need to be set before training can take place, namely the resolution of the image patch, which determines the number of potential haar wavelet weak classifiers  $f(x)$ , the number of training rounds  $M$ , the number of histogram bins  $h_b$  and the number of training example patches  $N$ . With 21x21 pixel image patches,  $N = 179,545$  ( 1205 positive patches, 178,340 negative patches),  $M = 200$ ,  $h_b = 25$  the training for each patch completes in  $\sim 20$ hrs on a single node of a 64bit multi-processor cluster running Linux. The feature models are trained independently therefore the whole model can be built in  $\sim 20$ hrs, if enough nodes are available. The parameters above are unlikely to be optimal. For example, it may well be possible to improve the results, by increasing the number of training rounds  $M$  or increasing the size of the training set, which currently only consists of 1205 face images (see Figure 1).

### 3.3 Boosted Feature Regression

In the feature detection approach described in Section 3.2 the model is trained on positive examples centred on a small neighbourhood around manually labelled feature locations. Negative examples are feature patches displaced from the true locations (see Figure 2).

However an obvious problem with feature detector training is where to draw the boundary between positive examples and nearby false examples. In Section 3.2 we take a conservative approach and only treat image patches centre on the true feature as positive examples. Patches between 1 pixel and 3 pixels away are treated as ambiguous, while patches greater than 5 pixels away are classed as false positives (a similar approach is adopted in [6]).

This works reasonably well, but is arbitrary and also throws away potentially useful information, such as the distance of each patch from the true positive. An alternative technique which makes use of this information is regression, which learns the relationship between the displacement to the true feature location and the textural appearance of the local neighbourhood around each feature point.

---

#### Algorithm 2 Gentle Boost Training Algorithm - Regression [9]

---

1. Start with input values  $x_i$  and target values  $y_i$  for  $i = 1, 2, \dots, N$  and  $F(x) = 0$  and small positive constant  $\alpha$ .
  2. Repeat for  $m = 1, 2, \dots, M$ :
    - (a) Fit the regression function  $f_m(x)$  by least squares of  $y_i$  to  $x_i$ .
    - (b) Select the  $f_m(x)$  with least error  $\sum_{i=1}^N (y_i - f_m(x_i))^2$
    - (c) Update  $F(x) \leftarrow \alpha F(x) + f_m(x)$
    - (d) Update the residual target value  $y_i \leftarrow y_i - f_m(x_i)$
  3. Output the regression function  $F(x) = \sum_{m=1}^M f_m(x)$
- 

We use the GentleBoost logistic regression method described by Friedman *et al.* [9]

(see Algorithm 2) which has very similar form to the GentleBoost classification training method use in Section 3.2. The same image patches  $x_i$  are used as in the classification training, however instead of  $y_i$  being class labels  $\{-1, +1\}$  they are local displacement values in the training image frame (suitably scaled - see Figure 3). The regression training therefore uses the whole training data available, whilst the classification training discards some ambiguous patches (marked with an X in Figure 3).










Image									
Regression	-4	-3	-2	-1	0	+1	+2	+3	+4
Class	-1	-1	X	X	+1	X	X	-1	-1

Figure 3: Examples of training patches for right eye from one of the training images (depicting translation in the x-coordinate). Regression training values are the displacement from the centre of the patch to the true eye pupil location (shown by a white cross). Classification training values are -1 for negative examples, +1 for positive examples, images marked with a X are ignored during classifier training

The GentleBoost regression algorithm then proceeds as described in Algorithm 2. The Haar wavelet functions  $f(x)$  are fitted to the weighted training patches  $x_i$  with displacement  $y_i$ . A function  $f(x)$  is selected at each stage, the residual displacements  $y_i$  are adjusted and after  $M$  rounds a strong regressor function  $F(x)$  is output.

Note that in Algorithm 1 the weights on each training example  $w_i$  are updated between training rounds. In Algorithm 2 the target values  $y_i$  vary between boosting rounds and the training examples have equal weight. Another important difference between the classification algorithm and the regression method is that the regression requires two models per feature point to predict the x and y displacements for each patch. The training time for each regression model is also increased slightly due to the extra training samples close to the true feature points being included in the training set (which are discarded during classifier training).

An additional parameter of the regression training algorithm is  $\alpha$  which represents the learning rate. This can be any value in the approximate range  $[0.1, 1.0]$  and needs to be chosen apriori. The value can be shown to be equivalent to the shrinkage parameter in Lasso Regression [9]. Small values of  $\alpha$  result in slower training, but more diverse feature selection. We set  $\alpha = 0.25$  in our experiments.

### 3.4 Summary of Method

At run-time the search proceeds as follows:-

1. Find initial feature points - for example using a global detection method
2. Iterate the following:-
  - (a) Search around the current feature location with a feature detector - Or alternatively predict the improved feature location using boosted regression
  - (b) Fit the shape model to the current set of feature locations to remove outliers

Until Converged.

## 4 Experiments

### 4.1 Test Criteria

The models described in Section 3 are applied to two publicly available test sets, with manually labelled ground truth, namely the BIOID [10] and XM2VTS [13] data sets. The criteria for success is the distance of the points computed using automated methods compared to manually labelled ground truth. The distance metric is shown in Equation 3.

$$m_e = \frac{1}{ns} \sum_{i=1}^n d_i \quad (3)$$

Here  $d_i$  are the Euclidean point to point errors for each individual feature location and  $s$  is the ground truth inter-ocular distance between the left and right eye pupils.  $n = 17$  as only the internal feature locations around the eyes, nose and mouth are used to compute the distance measure. The five feature points on the edge of the face (see Figure 1) are ignored for evaluation purposes, due to their high variability between different human annotators.

### 4.2 Full Search Results

The fully automatic search is investigated on the two faces test sets. Three separate procedures are investigated as follows:-

- AVG - Average points within the global Viola and Jones face detector (dashed line)
- Det-ASM - Detection Features and Active Shape Model, initialised with the average points (dotted line)
- Reg-ASM - Regression Features and Active Shape Model, initialised with the average points (solid line)

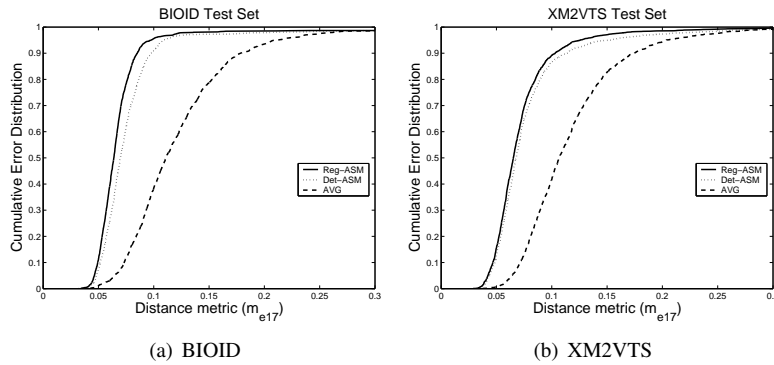


Figure 4: Cumulative distribution of point to point error measure on XM2VTS and BIOID test sets when using face detection to initialise the local search

Figure 4 shows that the Reg-ASM and Det-ASM give similar results on both the BIOID and XM2VTS data sets. Both local search methods combined with the ASM give

a large improvement relative to the average points found within the global face detector window. For example on the BIOID data set (see Figure 4(a) - dashed line) 75% of faces have a point to point error  $m_e < 0.15$  using the average points. However after the local ASM search is applied 95% of faces are found at this accuracy limit (see solid line). The Reg-ASM method performs slightly better than the Det-ASM on both data sets (compared solid+dotted lines in Figure 4).

The results in Figure 4 are comparable with the authors previous results published on the two data sets. For example using the same error measure on the BIOID data set the Constrained Local Model (CLM) algorithm [4] gives a 90% success rate at  $m_e < 0.1$  compared to 95% using the proposed Reg-ASM algorithm. For lower values of  $m_e$  the CLM is more accurate, however the Reg-ASM and Det-ASM methods described here are initialised using the average points from the face detector. In [4] the Pictorial Structure Matching (PSM) algorithm [7] is used as part of a three stage method. The Reg-ASM local search is also much more efficient than the CLM algorithm (see Section 4.4).

### 4.3 Displacement Results

In order to determine the range of convergence of the Reg-ASM and Det-ASM the tracking methods are systematically displaced from the true feature locations in the eight possible compass directions, by a percentage of the inter-ocular distance and the shape reset to the mean of the statistical shape model.

This gives a total of 8 starting search locations per image, to start the Reg-ASM and Det-ASM algorithms, at each of five possible displacements of 10%, 20%, 30%, 40% and 50% of the inter-ocular distance. The rate of convergence for the Reg-ASM and Det-ASM given a point to point error limit of  $m_e < 0.15$ , for this range of displacements is shown in Figure 5.

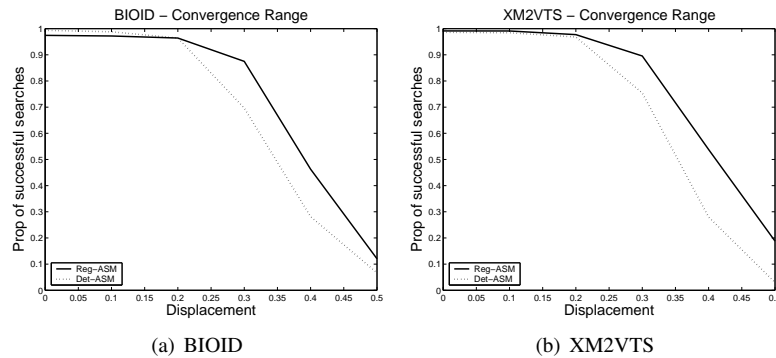


Figure 5: Range of convergence for regression and detection methods on XM2VTS and BIOID test sets

Figure 5 shows that the Reg-ASM has a wider range of convergence compared to the Det-ASM on both the BIOID and XM2VTS data sets. This is possibly due to the regression prediction for each point (in the Reg-ASM) being able to jump over false minima which may be found by the Det-ASM search. However both algorithms have some in-built ability to avoid false minima due to the ASM shape fitting step which removes outlying predictions for individual feature points.



## 4.4 Timings

The local search time using the Reg-ASM and the Det-ASM methods is dependent on the search image and the starting displacement. However both algorithms converge in fewer than 5 iterations in most cases. Therefore the search speed is dependent on the time for one iteration of the ASM.

In our implementation one iteration of the Reg-ASM takes  $\sim 3$ ms compared to  $\sim 25$ ms with the Det-ASM<sup>1</sup>, using a C++ implementation on a P4 3GHz processor. Therefore the Reg-ASM is approximately eight times quicker than the Det-ASM. If 1-5 iterations are required when tracking a face with the Reg-ASM in a video sequence the frame rate will be approximately 60-300 frames per second.

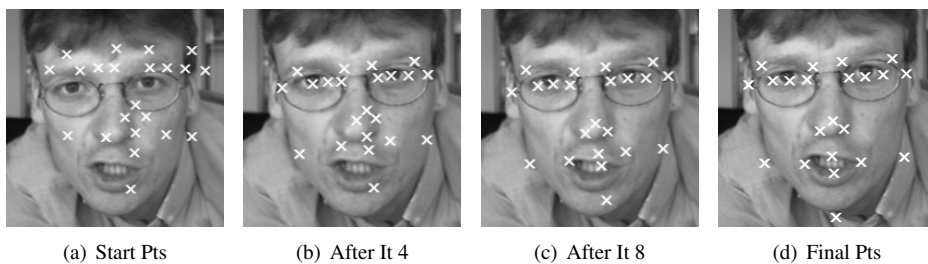


Figure 6: Example of Iterative Reg-ASM Search on BIODID image

## 5 Conclusions

We have compared two local feature updates methods within the Active Shape Model framework. The boosted regression approach is shown to have a wider range of convergence compared to the boosted classifier method on two publicly available face data sets. The boosted regression method is also more computationally efficient by a factor of eight, which makes it suitable for use in real time systems.

Future work will involve building larger models with more data and different data sets. We are particularly interested in applying the boosted regression approach to high dimensional medical images, as in more than two dimensions the feature detection search at run-time becomes prohibitively expensive. We may also apply the regression update step in other formulations such as the AAM.

The boosted regression feature prediction method described is an extremely efficient local search algorithm ( $> 60$  frames per second), which improves on standard boosted feature detection approaches. We anticipate that this form of boosted regression update will be useful in other areas of computer vision.

<sup>1</sup>Note it may be possible to improve the efficiency of the Det-ASM by introducing a cascade structure for each classifier as in [17]. However the fact that the classifier has to search the local neighbourhood will always make it slower than the regression model, if both methods use the same number of weak learners.

## References

- [1] T. Ahonen, A. Hadid, and M. Pietikainen. Face recognition with local binary patterns. In *8<sup>th</sup> European Conference on Computer Vision 2004, Prague, Czech Republic*, pages 469–481, 2004.
- [2] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *5<sup>th</sup> European Conference on Computer Vision 1998, Freiburg, Germany*, volume 2, pages 484–498, 1998.
- [3] T. F. Cootes and C. J. Taylor. Active shape models. In *3<sup>rd</sup> British Machine Vision Conference 1992*, pages 266–275, 1992.
- [4] D. Cristinacce and T. Cootes. Detection and tracking with constrained local models. In *17<sup>th</sup> British Machine Vision Conference 2006, Edinburgh, Scotland*, pages 929–938, 2006.
- [5] N. Dowson and R. Bowden. Simultaneous modeling and tracking (smat) of feature sets. In *23<sup>rd</sup> Computer Vision and Pattern Recognition Conference 2005, San Diego, USA*, pages 99–105, 2005.
- [6] M. Everingham and A. Zisserman. Regression and classification approaches to eye localisation in face images. In *7<sup>th</sup> International Conference on Automatic Face and Gesture Recognition 2006, Southampton, UK*, pages 441–446, 2006.
- [7] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61:55–79, 2005.
- [8] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.
- [9] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28:337–407, 2000.
- [10] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. Robust face detection using the hausdorff distance. In *3<sup>rd</sup> International Conference on Audio- and Video-Based Biometric Person Authentication 2001, Halmstad, Sweden*, pages 90–95, 2001.
- [11] G. Langs, P. Peloschek, R. Donnerer, M. Reiter, and H. Bischof. Active feature models. In *18<sup>th</sup> International Conference on Pattern Recognition 2006, Hong Kong, China*, pages 417–420, 2006.
- [12] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *IEEE International Conference on Image Processing*, pages 900–903, New York, USA, 2002.
- [13] K. Messer, J. Matas, J. Kittler, J. Luetin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *2<sup>nd</sup> International Conference on Audio- and Video-Based Biometric Person Authentication 1999, Washington DC, USA*, pages 72–77, 1999.
- [14] M. Seise, S. McKenna, I. W. Ricketts, and C. A. Wigderowitz. Learning active shape models for bifurcating contours. *IEEE Transactions on Medical Imaging*, 26(5):666–677, 2007.
- [15] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
- [16] B. van Ginneken, A. F. Frangi, J. J. Staal, B. M. ter Haar Romeny, and M. A. Viergever. Active shape model segmentation with optimal features. *IEEE Transactions on Medical Imaging*, 21:24–933, 2002.
- [17] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *19<sup>th</sup> Computer Vision and Pattern Recognition Conference 2001, Hawaii, USA*, volume 1, pages 511–518, Kauai, Hawaii, 2001.
- [18] M. Wimmer, F. Stulp, S. Tschechne, and B. Radig. Learning robust objective functions for model fitting in image understanding applications. In *17<sup>th</sup> British Machine Vision Conference 2006, Edinburgh, Scotland*, pages 1159–1168, 2006.
- [19] Y. Zheng, X. S. Zhou, B. Georgescu, S. K. Zhou, and D. Comaniciu. Example based non-rigid shape detection. In *9<sup>th</sup> European Conference on Computer Vision 2006, Graz, Austria*, pages 423–436, 2006.