Facial feature detection using AdaBoost with shape constraints

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

Abstract

Recently a fast and efficient face detection method has been devised [11], which relies on the AdaBoost algorithm and a set of Haar Wavelet like features. A natural extension of this approach is to use the same technique to locate individual features within the face region. However, we find that there is insufficient local structure to reliably locate each feature in every image, and thus local models can give many false positive responses. We demonstrate that the performance of such feature detectors can be significantly improved by using global shape constraints. We describe an algorithm capable of accurately and reliably detecting facial features and present quantitative results on both high and low resolution image sets.

1 Introduction

This paper addresses the problem of locating facial features (eyes, nose, mouth corners and so on) in images of frontal faces. Locating such features is an important stage in many facial image interpretation tasks (such as face verification, face tracking or face expression recognition). We adopt the fast and efficient face finder recently described by Viola and Jones [11] to locate the approximate position of each face in an image. We then use the same method, trained on regions around facial feature points, to locate interior points on the face. However, there is often insufficient local structure around each feature to train really reliable feature finders. We find that when set with thresholds sufficient to locate the true position reasonably frequently, such detectors produce many false positives. To select the most suitable candidates we use statistical models of the configurations of the points. We find that combining feature detectors with such statistical shape models gives a significant improvement in both the reliability and the overall accuracy of the feature detection system.

In the following we describe the approach in more detail, and demonstrate its application to finding features in two data sets. Though demonstrated on faces, the approach is clearly applicable to a wide variety of image interpretation tasks.

2 Background

Face detection has received much attention within the computer vision community, see [7] for a survey or Viola [11] which describes the fast and efficient face detection method used in this paper.

BMVC 2003 doi:10.5244/C.17.24

Facial feature detection methods generally model two types of information. The first is local texture around a given feature, for example the pixel values in a small region around an eye. The second is the geometric configuration of a given set of facial features, e.g. both eyes, the nose, mouth etc. Many different methods of modelling this shape and texture information present in the human face have been proposed.

Burl *et. al.*[1] use multi-oriented, multi-scale Gaussian derivative filters to model the texture around key points on the face. The shape is modelled using the shape statistics of Dryden and Mardia [3]. The most likely shape, found by combining the feature detections, is declared the best match to the face. Therefore the likelihood of each feature detection is ignored when comparing two possible feature point configurations, only the shape information is used.

A well known approach due to Wiskott *et. al.*[13] uses ”Gabor Jet” feature detectors and models the distribution of facial features with a graph structure. The quality of fit function for such a model has two parts. The first is the sum of the Gabor jet detector responses. The second is the similarity between the inter-features distances compared to the graph model. These two metrics are combined with a manually adjusted weighting. Hence shape and feature detection are combined in an arbitrary way.

A faster and simpler method is due to Feris *et. al.*[9]. Here a two stage hierarchy of Gabor wavelet networks is used. The first localises the whole face, the second stage searches for individual features. The approximate shape of the face predicted from the first stage is used to guide the feature detection, hence shape is not modelled explicitly and only local image structure is used.

Other methods for facial feature detection have been proposed. For example, neural networks [10], steerable filters [4] and colour space methods [6]. However, none of these method use shape, which we believe to be an integral part of facial feature detection and feature detection in general. Knowledge of face shape is needed to provide robustness to any system, because local feature detectors are inherently noisy.

Cootes *et. al.*[2] combine shape and texture modelling in the active appearance model (AAM) approach to face matching. The shape and texture are combined in a PCA space. The model searches a new image iteratively by using the shape and texture to drive the model parameters. Given a good enough initialisation the AAM converges to the correct solution, but otherwise is prone to local minima. The method outlined in this paper could be used to seed the AAM.

The method proposed in this paper utilises the successful feature detection method due to Viola [11] combined with the statistical shape models of Dryden and Mardia [3]. Local AdaBoost templates are used to localise each feature that are swift to compute. However, used individually these feature detectors are shown to be insufficient. A shape model must be used to constrain the configuration of a set of candidate feature points. An efficient search algorithm is introduced that allows the facial feature detections to be grouped and tested. This method avoids local minima, by searching many possible feature point configurations. The method allows for feature detector failures by predicting the positions of missing features using the shape model.

3 Methodology

3.1 Face detection

To build a system capable of automatically labelling features on the face it is first necessary to localise the face in the image. We apply the recent method due to Viola and Jones [11].

The Viola-Jones Detector consists of three parts. The first is an efficient method of encoding the image data known as an ”integral image”. This allows the sum of pixel responses within a given sub-rectangle of an image to be computed quickly and is vital to the speed of the Viola-Jones Detector . The second element is the application of a boosting algorithm known as AdaBoost [5] to select appropriate features that can form a template to model human face variation. The third part is a cascade of classifiers that speeds up the search by quickly eliminating unlikely face regions.

We implemented the full Viola-Jones Detector to localise the face in the image. The output of the face detector is a image region containing the face, which is then examined to predict the location of the internal face features. The facial feature detection method described here uses individual feature patch templates to detect points on the face. These feature models are described in more detail in section 3.2. For efficiency, the same integral image is used for both the initial face detection stage and the subsequent facial feature detection.

3.2 Feature Point Models

The facial feature models are constructed using the same method as each individual level of the Viola-Jones Detector cascade. The method of building an AdaBoost template from simple Haar wavelet like features is described by Viola and Jones [12]. The method requires a set of positive and negative image regions. In the work of Viola and Jones the positive examples are human faces and the negative examples are regions known not to contain a human face. In this paper the positive examples are image patches centred on a particular facial feature and the negative examples are image patches randomly displaced a small distance from the same facial feature.

Note that when building a face template (for face detection) using this method and classification problems in general, a major problem is producing a representative training set. The space of possible faces is extremely large, however the space of all possible non-face images is even larger. This problem is solved by Viola and Jones by using false positives from the earlier versions of the cascade. However with local feature point models the non-feature texture is simply modelled by the region surrounding the given feature.

Our training set consists of 995 faces with manual annotated landmarks for key facial features, some examples are shown in figure 1. Note the training set contains a significant amount of head rotation and examples of closed eyes. Some examples image patches used to build the right eye model are shown in figure 2.

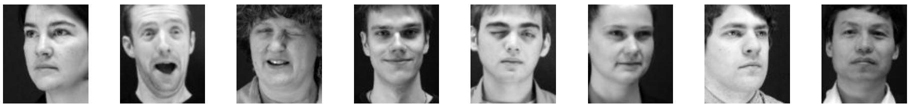


Figure 1: Example training images

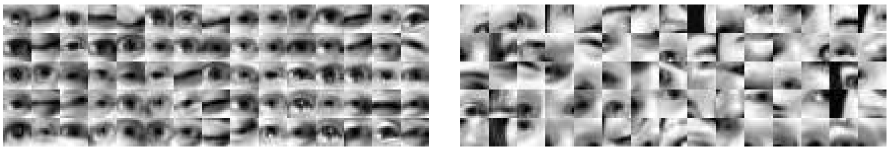


Figure 2: The positive and negative training examples for the right eye feature model

Local models are built for four features on the face namely the left and right eyes and the left and right corners of the mouth. The first few features selected by AdaBoost for the right eye feature model are shown in figure 3.

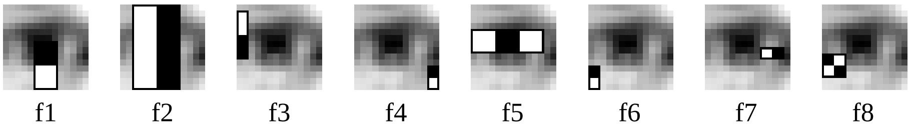


Figure 3: Features selected by AdaBoost, overlaid on an example from the training set

Additionally the probability of each feature response is approximated, by learning the distribution of responses to both positive and negative examples. The probability of a given feature response, r, matching the correct point within the search window, $P(F|r)$, is given by equation 1.

$$P(F|r) \quad = \quad \frac{p(r|F)P(F)}{p(r|F)P(F)+(1-P(F))p(r|B)} \qquad (1)$$

Here $p(r|F)$ is the probability density of a given response at the true feature location, this is estimated by creating a histogram of responses to a positive verification set. $p(r|B)$ is the probability density of a given response away from the true feature location and is similarly estimated using a histogram learnt from a negative verification set. $P(F)$ is the prior probability of a correct feature match and is estimated as $P(F) = 1/n$, where n is the number of points evaluated by the feature detector within the search region.

For each feature detector a suitable threshold T_f is learnt from a verification set, so if $P(F|r) > T_f$, a feature candidate is accepted. The discriminating power of these local feature detectors is investigated in section 5.2.

3.3 Shape Constraints

A set of candidate feature points is tested using shape constraints in two ways. Firstly a shape model is fitted to the set of points and the likelihood of the shape assessed. Secondly limits are set on the orientation, scale and position of a set of candidate feature points relative to the orientation, scale and position implied by the global face detector.

The shape model is built using methods developed by Dryden and Mardia [3] and fitted to a set of unseen points using a least squares fit algorithm. Following [3] we assume that after aligning the points into a common co-ordinate frame, the distribution is a multi-variate Gaussian, the parameters of which can be estimated from the training set. Thus for any new points we can estimate the probability density of a given shape $p_s(\mathbf{x})$. A suitable threshold T_s is learnt from a verification set and used to distinguish plausible from implausible configurations, so if $p_s(\mathbf{x}) > T_s$, the shape is accepted, otherwise it is discarded.

To fit a shape model to a set of points it is also necessary to compute the transformation (i.e. orientation, scale and translation) from the model frame to a given set of candidate points. The transformation to the mean points predicted by the global search is also computed. These two transformations are compared and limits imposed on the scale difference s_{max}, angle difference θ_{max} and translations x_{max} and y_{max}. The limits are set with knowledge of the accuracy of the global search. In this paper the limits used are $s_{max} = 10\%$, $\theta_{max} = 10^o$ and $x_{max} = y_{max} = 10\%$ of the global search region.

If any of the constraints are broken then the given set of candidate points is rejected and will not be accepted as a face candidate.

3.4 Shape guided feature detection

We analyse the training set to determine the range of variation in position of the features relative to the bounding box found by the full face detector. During search, having found the full face region we scan each relevant sub-region for plausible candidates. The feature detectors each return a list of candidate points that pass a threshold T_f on the probability of a correct match $P(F|r)$. The list of candidate points need to be combined into face candidates and the best face candidate selected.

A simple method of selecting the best combination of points is to threshold the detector responses, form all possible candidates and select the face candidate with the highest set of feature responses $P(F|r)$ that also passes the constraints imposed by the shape model. However, an obvious problem with this approach is the combinatorial explosion that results if too many candidate points are returned for too many feature points. For example with 7 feature points each returning 9 candidate locations and allowing for missing features, the number of possible face candidates is $(9+1)^7 = 10^7$. It is infeasible to evaluate this number of candidates in a sensible amount of time.

Algorithm 1 (Shape guided search)

1. *Order the feature candidates for each feature detector, largest response first.*
2. *Set* $i = 1$
3. *Restrict the search for a face candidate to the best i responses of each detector.*
4. *If a feature detector has less than i point candidates then it is allowed a wildcard.*
5. *Ensure that each face candidate contains at least one candidate point that is the* i^{th} *response of a detector* [1]
6. *Form all face candidates and test each with the shape model.*
7. *If no face candidate passes the shape constraint set* $i = i + 1$ *and go to step 3.*
8. *Rank all face candidates that pass the shape constraint, by the number of detected feature points and then by the probability* p_{set} *of the set of feature points. Here* $p_{set} = \prod_x P(F|r_x)$ *for each feature response* r_x. [2]
9. *If the best candidate has missing features these are predicted using the shape model.*

[1] This ensures that point sets from previous iterations of i are not re-evaluated

[2] Note this ranking scheme means that a candidates set with 4 points is always ranked higher than any candidate with only 3 points

We wish to find the best candidate, using the feature responses and the shape constraints, but also limit the combinatorial explosion. This is achieved in three ways. The first is simply by limiting the number of feature points n_p. The second is by restricting the number of points any one feature detector can return n_{max}. The third method is to employ an efficient search method as described in algorithm 1.

The aim of algorithm 1 is to select the highest scoring candidate with the maximum number of feature points that satisfy the shape constraint. The algorithm will always find the best candidate for a given value of i, i.e. when allowed to select from the first i point candidates of each detector. The procedure makes searching tractable by ignoring weaker point candidates and returns the same candidates as a full search.

The main parameters used in algorithm 1 are the number of feature points n_p and the maximum number of candidate points retained by each feature detector n_{max}. In this paper, $n_p = 4$ and the features used are the centre of both eyes and the corners of the mouth, while n_{max}=5.

These parameters mean that a point set must be found from the top five point candidate returned by each feature detector. If a feature detector returns less than 5 point candidates, because the probability of any correct match is too low, then a wildcard will eventually be used for that feature. In practice these values for n_p and n_{max} give good results and reduce the processing time relative to a full search of all possible point candidates.

4 Experiments

There are two data sets used to test the feature detection algorithm. The first is a subset of the XM2VTS data set [8], consisting of 1817 frontal faces taken from photo sessions 1-4. The images are taken under controlled conditions against a flat background, but the data set does contain people with facial hair and wearing glasses. The second data set is a smaller set (212 images) taken with a cheap web camera in our lab. The images are low quality and contain background clutter, making the face detection task much more challenging than with the XM2VTS data. Also, there is more head rotation which makes the task of facial feature detection more difficult. Example images from both data sets are shown in figure 4.

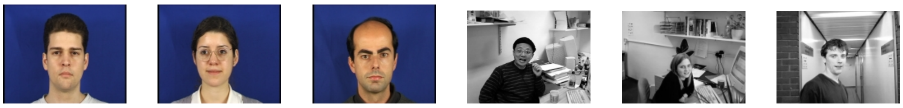


Figure 4: Example test set images

We apply the global search to each test image and return the highest ranking global candidate. We consider the global search to be successful if the mean distance from the found eye positions (as predicted by the global face template) to the true eye positions (annotated manually) is less than 30% of the true inter-occular distance. To concentrate on the feature detection, we discard those examples in which the global search failed this threshold. Global search failed on eleven of the XM2VTS images (11/1817=0.6%) and twenty of the webcam images (20/212=9.4%), which were not included in the analysis of feature detection performance below.

When evaluating the feature search, we measure the positional error of each feature and express it as a proportion of the inter-occular distance.

5 Results

5.1 FROC curve for individual feature detectors

Figure 5 shows the free receiver operator characteristic (FROC) curve, plotting true positive rate against number of false positives per region, for eye and mouth detectors on the XM2VTS data. It demonstrates that the eye centres are more reliably found than the mouth corners. All four detectors are noisy, and individually are not sufficiently reliable. However, by constraining the search, more accurate detections can be made.

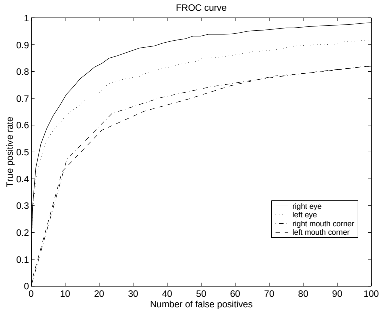


Figure 5: FROC curves for the individual feature point detectors, tested on a subset of the XM2VTS database

5.2 Facial feature detection accuracy

For each image we can calculate the mean positional error for all features, m_e. Figures 6(a), 6(b) show the cumulative probability distributions for m_e for the XM2VTS and webcam data sets respectively. We show results for three different methods of feature detection, as follows.

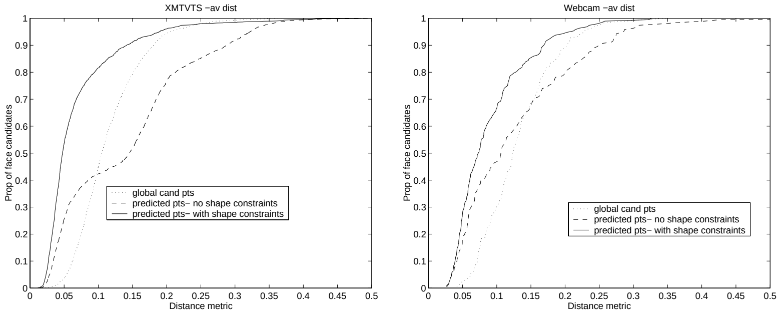
1. Mean position predicted from full face match (no feature search) (dotted line).

2. Most probable feature response in each region(no shape constraints) (dashed line)

3. Best combination of features using shape constraints (solid line).

The results on the XM2VTS data set (Figure 6(a)) show that the raw feature search actually gives worse results than assuming the features are in their mean positions. However, using shape constraints to select the best combination of features leads to significantly better performance for almost all cases.

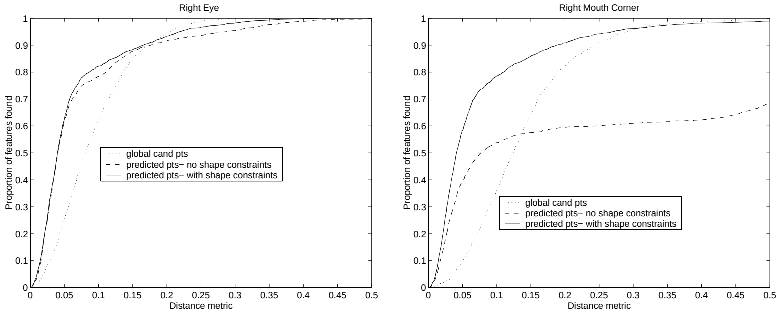
The reasons for the success of the shape constrained method and the poor performance of the unconstrained search can be investigated by examining the proportion of successful searches for individual features. Figures 7(a) and 7(b) show the cumulative distribution of errors for the right eye and right mouth corner respectively.



(a) XM2VTS Data Set

(b) Webcam Data Set

Figure 6: Average mean positional error(m_e) of all 4 features



(a) Right Eye

(b) Right Mouth Corner

Figure 7: Detection accuracy for individual eye and mouth features on the XM2VTS database

These graphs demonstrate that the eye position is better localised using feature search than raw prediction from the mean, but that for eyes the additional shape constraints do not make a significant difference. However, with mouth corners the individual feature search is poor, but is greatly improved by the use of shape constraints.

Similar results have been obtained for the left eye and mouth corner, and for experiments on the webcam data, see figure 6(b). Example search results are shown in figure 8.

5.3 Speed of the feature detector

Timings were carried out on a set of 320*262 pixel images, using modest hardware, a 500Mhz PII processor. The speed of the system is dependent on the pixel values in the image, but can broadly be broken down as shown in table 1.

The time to find the best point candidate stage varies greatly, because it implements the search strategy outlined in algorithm 1. If the best feature responses form a legal shape

Global search	~ 250ms
Local feature search	~ 100ms
Find best point candidate set	$\sim 0-150$ms

Table 1: Search times using 500Mhz PII processor

then the search exits immediately, but otherwise more possibilities need to be investigated. The whole search time therefore varies 350-500ms i.e. < 0.5 secs.

6 Summary and Conclusions

The graphs of figure 6 show that shape constrained search significantly improves on the feature detection accuracy possible when using only the best response from each feature detector. For features that exhibit large variation in location and appearance on the face, such as the corners of the mouth, local image structure is not enough to allow reliable detection. The shape of a set of feature points must be used as a constraint to remove false matches.

A method is proposed (see algorithm 1) to search the space of possible feature point combinations in an efficient manner. After the shape constrained search has taken place on the XM2VTS data set, the average feature distance is within 5% of the eye separation for 65% of faces, 10% for 85% of faces and 15% for 90% of faces and the method improves the initial feature points predicted by the global search detector in 88.8% of cases. Similar results are observed when applying the search to more difficult images, which contain more head pose variation. The method is also quick requiring less than 0.5 secs for a 320*262 image on a 500Mhz PII processor.

In the above we have described results for only four feature points. We have performed further experiments with seven points (adding the nose and eyebrows), and obtained similar results in terms of the improvement obtained by using the shape constraints.

Figure 7(a) demonstrates that the shape constraints do not significantly improve the positional accuracy of the eyes location compared to simply selecting the best feature match. This suggests that the best strategy is to accept the best feature match for each eye, which will significantly reduce the complexity of the search for combinations of other candidate points. In further work we will investigate using alternative feature detectors and the use of a multi-resolution approach to obtain more accurate estimates of feature positions.

In conclusion, the method is fast, robust and reliable, and significantly improves on the feature candidates provided by the global search. Though demonstrated on faces, the approach is clearly applicable to a wide variety of image interpretation tasks.

References

[1] M. Burl, T. Leung, and P. Perona. Face localization via shape statistics. In 1^{st} *International Workshop on Automatic Face and Gesture Recognition 1995*, Zurich, Switzerland, 1995.

[2] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In H.Burkhardt and B. Neumann, editors, 5^{th} *European Conference on Computer Vision*, volume 2, pages 484–498. Springer, Berlin, 1998.

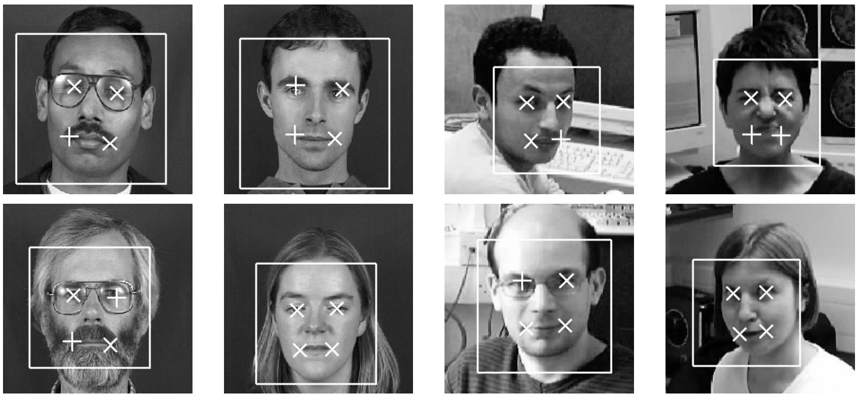


Figure 8: Example search results, x= detected point, += predicted point, box= global search region

[3] I. Dryden and K. V. Mardia. *The Statistical Analysis of Shape*. Wiley, London, 1998.

[4] R. Herpers et al. Edge and keypoint detection in facial regions. In 2^{nd} *International Conference on Automatic Face and Gesture Recognition 1996*, pages 212–217, Killington, Vermont, USA, October 1996.

[5] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *2nd European Conference on Computational Learning Theory*, 1995.

[6] H.P Graf, E.Casotto, and T. Ezzat. Face analysis fro synthesis of photo-realistic talking heads. In 4^{th} *International Conference on Automatic Face and Gesture Recognition 2000*, pages 189–194, Grenoble, France, 1995.

[7] E. Hjelmas and B.Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83:235–274, 2001.

[8] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. Xm2vtsdb: The extended m2vts database. In *Proc. 2nd Conf. on Audio and Video-based Biometric Personal Verification*. Springer Verlag, 1999.

[9] K. Toyama R. S. Feris, J. Gemmell and V.Krüger. Hierarchical wavelet networks for facial feature localization. In 5^{th} *International Conference on Automatic Face and Gesture Recognition 2002,* Washington D.C. , USA, May 2002.

[10] M.J.T Reinders, R.W.C Koch, and J.J Gerbrands. Locating facial features in image sequences using neural networks. In 2^{nd} *International Conference on Automatic Face and Gesture Recognition 1996*, pages 230–236, 1996.

[11] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition Conference 2001*, volume 1, pages 511–518, Kauai, Hawaii, 2001.

[12] P. Viola and M. Jones. Robust real-time object detection. In *In ICCV Workshop on Statistical and Computation Theories of Vision*, Vancouver, Canada, 2001.

[13] L. Wiskott, J.M. Fellous, N. Kruger, and C.von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.