Facial Feature Detection and Tracking with Automatic Template Selection

D. Cristinacce and T. F. Cootes
Dept. Imaging Science and Biomedical Engineering
University of Manchester
Manchester, M13 9PT, U.K
{david.cristinacce,tim.cootes}@manchester.ac.uk

Abstract

We describe an accurate and robust method of locating facial features. The method utilises a set of feature templates in conjunction with a shape constrained search technique. The current feature templates are correlated with the target image to generate a set of response surfaces. The parameters of a statistical shape model are optimised to maximise the sum of responses. Given the new feature locations the feature templates are updated using a nearest neighbour approach to select likely feature templates from the training set. We find that this Template Selection Tracker (TST) method outperforms previous approaches using fixed template feature detectors. It gives results similar to the more complex Active Appearance Model (AAM) algorithm on two publicly available static image sets and outperforms the AAM on a more challenging set of in-car face sequences.

1. Introduction

This paper describes a method of automatically finding facial features, such as eye pupils, corners of the mouth etc in both static images and video sequences. This is important for many tasks such as face recognition and automatic avatar generation.

Our Template Selection Tracker (TST) algorithm consists of a shape model and a training set of possible feature templates learnt from a set of manually labelled face images. The algorithm consists of two elements- a template selection step and a shape constrained search step.

Given current image points the template selection proceeds by fitting the shape model and using the Euclidean distance in the shape space to perform a nearest neighbour search of the training examples. The set of training examples selected by shape are then correlated with the current image texture to select the closest template texture from the training set.

The best matching feature templates are then applied to the search image using normalised correlation. This generates a set of response surfaces. The quality of fit of the model is optimised using the Nelder-Meade simplex algorithm [11] to drive the parameters of the shape model in order to maximise the sum of responses at each point. Given a new set of candidate feature locations the templates are regenerated and the search proceeds iteratively.

This Template Selection Tracker (TST) approach, summarised in Figure 2, is shown to be robust, relatively quick and provide superior tracking performance compared to the Active Appearance Model matching method [1].

2 Background

There are many examples of computer vision techniques that combine both shape and texture to build models and match to unseen images [1][2] [5][7].

One popular approach is the Active Appearance Model (AAM) algorithm [1], which uses a combined statistical model of shape and texture. The AAM searches by using the texture residual between the model and the target image to predict improved model parameters to obtain the best possible match. However in this paper we propose a nearest neighbour template search method, which uses the shape model as a constraint.

Given a set of region templates the TST search proceeds in a similar manner to the author's previous work [3]. However in [3] the feature templates are fixed during search, whilst here we propose an iterative scheme which selects appropriate templates given the current feature points and target image. Also in [3] the shape model parameters were constrained using hard limits, here we adopt a soft penalty term based on the log-likelihood of the shape.

The search method is also related to the earlier Active Shape Model (ASM) algorithm [2], however the ASM also uses fixed templates and only uses the shape model to update the feature locations after computing the best match of each detector. Our approach utilises the whole response

Figure 1. Training Examples

surface, allowing a better balance between shape and feature response.

Our algorithm is similar to the recent SMAT algorithm described by Dowson and Bowden [5]. The SMAT method tracks an object given an initialisation and then uses mutual information to select the closest matching feature templates, from a clustered set of templates sampled from previous frames and also updates a shape model of the feature configurations. The SMAT model is continuously updated, so therefore (assuming successful tracking) becomes more robust overtime. The TST uses normalised correlation to select templates, learnt from a fixed training set. In comparison the TST is not prone to failure in early frames, can be applied to search static images, not just video and the offline model is guaranteed to contain no false examples, but at the expense of requiring a manually labelled training set.

An elegant method of combining feature responses and shape constraints is due to Felzenswalb [7]. This Pictorial Structure Matching (PSM) approach is very efficient due to the use of pairwise constraints and a tree structure. However PSM is mainly a global search method and does not use a full shape model.

In this paper we use the Viola and Jones [13] face detector to find the face in the image. Within the detected face region we apply smaller Viola and Jones feature detectors constrained using the PSM algorithm [7], to compute initial feature points. We then refine these feature points using our TST local search and compare the results with the AAM algorithm [1].

3 Methodology

3.1 Shape Modelling

To build a joint shape and texture model we require a training set of images with corresponding labels for each feature. Figure 1 shows our training set which consists of 1052 manually labelled images.

A statistical shape model is built from the training set using the method of Cootes *et al.* [1]. This provides a paramaterisation, $\mathbf{b}$, of shapes similar to the training set,

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b} \qquad \mathbf{b} = \mathbf{P}^T(\mathbf{x} - \bar{\mathbf{x}}) \tag{1}$$

Where $\bar{\mathbf{x}}$ is the mean shape, $\mathbf{P}$ is a set of orthogonal modes of variation and $\mathbf{b}$ is a set of shape parameters. The shape model parameter vector $\mathbf{b}$ can be estimated for new shapes using the transpose matrix $\mathbf{P}^T$.

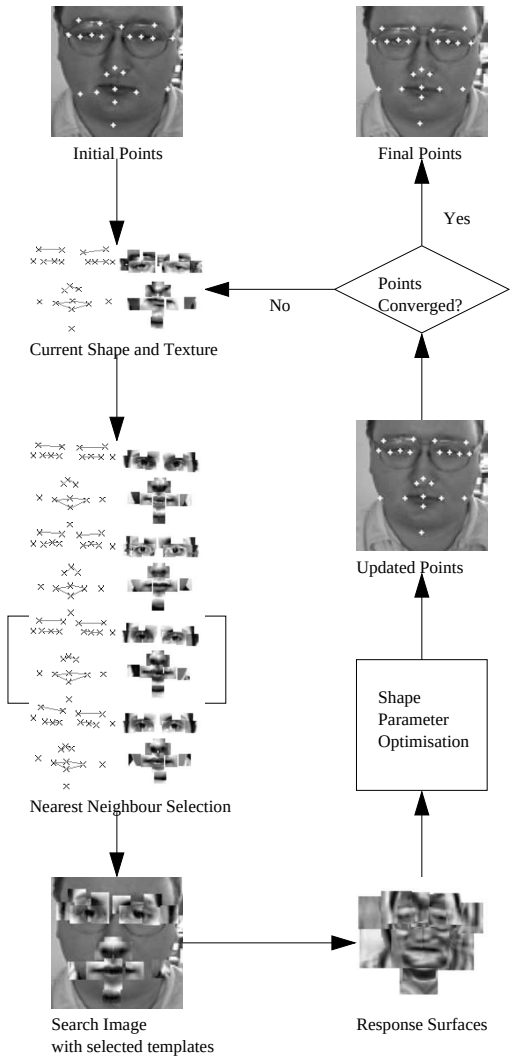


Figure 2. Overview of Template Selection Tracker (TST) algorithm

3.2 Nearest Neighbour Template Model

The labelled training set in Figure 1 allows a rectangular training patch to be sampled around each feature (normalised for scale changes). A set of these feature templates is computed for each training image and stored along with the corresponding shape model parameters $\mathbf{b}$.

Given an unseen image and approximate feature points a set of likely feature templates can now be generated using an efficient nearest neighbour search. The shape model is fitted to the current feature points to obtain the current

shape parameters, $\mathbf{b}$. These parameters are then compared with the stored training example shapes and the K closest matching shapes selected using the Euclidean distance in shape space. The top K training examples selected by shape matching are then tested by comparing with the texture sampled from the current image using normalised correlation. The best matching training example textures are then used to form detectors for each facial feature.

Given initial feature points, some of which lie close to the true feature location this simple nearest neighbour search will generate a likely set of templates, close to the true orientation of the current face. The only parameter that needs to be set is K, the initial number of shape matches retained. For our training set (see Figure 1) which consists of 1052 images we find K=20 provides good results.

3.3 Shape Constrained Template Search

The templates generated using the method described in Section 3.2 are used to improve the localisation accuracy of the feature points. The feature detectors are applied to the current image, to compute a set of response images (one for each feature). Let (X_i, Y_i) be the position of feature point i and $I_i(X_i, Y_i)$ be the response of the i_{th} feature template at that point. The positions can be concatenated into a vector $\mathbf{X}$,

$$\mathbf{X} = (X_1, \ldots, X_n, Y_1, \ldots, Y_n)^T \tag{2}$$

Where $\mathbf{X}$ is computed from the shape parameters $\mathbf{b}$ and a similarity transformation T_t from the shape model frame to the response image frame. $\mathbf{X}$ is calculated as follows.

$$\mathbf{X} \approx T_{\mathbf{t}} \left(\bar{\mathbf{x}} + \mathbf{P}\mathbf{b} \right) \tag{3}$$

The parameters of the similarity transform, $T_{\mathbf{t}}$ and, shape parameters $\mathbf{b}$ are concatenated into $\mathbf{p} = \left(\mathbf{t}^T | \mathbf{b}^T\right)^T$. Therefore $\mathbf{X}$ can be represented as a function of $\mathbf{p}$. Given a starting value for $\mathbf{p}$ the search proceeds by optimising a function $f(\mathbf{p})$ based on the image response surfaces I_i and the statistical shape model learnt from the training set. The objective function we use is

$$f(\mathbf{p}) \quad = \quad \sum_{i=1}^{n} I_i(X_i, Y_i) + R \sum_{j=1}^{s} \frac{-b_j^2}{\lambda_j} \tag{4}$$

The second term is an estimate of the log-likelihood of the shape given shape parameters b_j and eigenvalues λ_j. This log-likelihood follows the approach of Dryden [6] in assuming the b_j are independent and Gaussian distributed. The parameter R is a weight determining the relative importance of good shape and high feature responses. The value of R can be determined by computing the ratio of $\sum_{i=1}^{n} I_i(X_i, Y_i)$ and $\sum_{j=1}^{s} \frac{b_j^2}{\lambda_j}$ when applied to a verification set with human labelled ground truth. The optimisation of $f(\mathbf{p})$ is performed using the Nelder-Meade simplex algorithm [11].

Equation 4 differs from the objective function used in the author's previous work [3], which uses hard limits on the shape parameters b_j. Using hard limits avoids the need for a trade off between shape and feature responses, but is also capable of allowing some unlikely shapes (e.g. if all the shape parameters b_j are close to their individual limits).

Another refinement we introduce is the use of distance transforms to smooth the response surfaces. The distance transform used is similar to that described by Felzenswalb *et al.* [7], who smooth feature detector responses to allow for deformable shape variation when performing a global image search. The smoothing of the response images helps the optimisation avoid false minima, but also accounts for the residual variation of the shape model. The cost per unit distance in the response surface frame can be determined from the residual eigenvalues $\lambda_j (j > s)$ discarded when building the statistical shape model.

3.4 Template Selection Tracker Algrorithm

The full TST search algorithm is shown in Figure 2 and combines the methods described in Sections 3.2 and 3.3. The procedure is as follows:-

1. Input an initial set of feature points.
2. Repeat:-
   (a) Use nearest neighbour matching to select the closest texture from the training set (see Section 3.2).
   (b) Use the shape constrained search method (see Section 3.3) to predict a new set of feature points.

   Until Converged.

When tracking the initial points are propagated from the previous frame. On a new sequence (or if tracking failed) a global search can be used.

4 Experiments

4.1 Test Data

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 5.

$$m_e \quad = \quad \frac{1}{ns} \sum_{i=1}^{i=n} d_i \tag{5}$$

Here d_i are the 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.

4.2 Detection Experiments

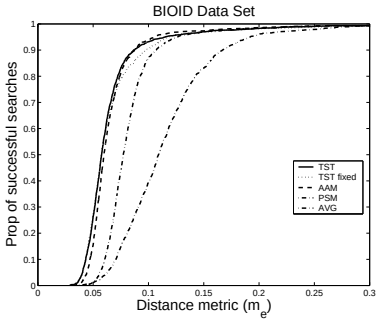
The localisation accuracy of the TST and AAM algorithms is tested by applying the methods to the publicly available BIOID [8] and XM2VTS [10] data sets. Note that these images are completely independent of the training images which contains different people imaged under different conditions (see Figure 1).

Our procedure for finding initial facial feature locations in a static image is to apply the Viola and Jones face detector [13], then apply similar smaller region detectors within the face candidate region, which are constrained using the Pictorial Structure Matching (PSM) approach due to Felzenswalb [7]. This method produces a set of points from which to initialise the TST and AAM algorithms. Five different procedures are evaluated as follows:-

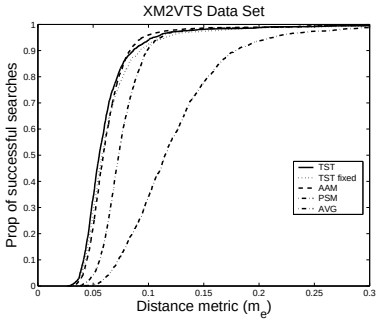
- AVG - Average points within the global Viola and Jones face detector.
- PSM - Pictorial Structure Matching points found within the Viola and Jones candidate face region.
- AAM - Active Appearance Model algorithm initialised with PSM points [1].
- TST fixed - Template Selection Tracker initialised with the PSM points, but restricted to fixed templates (mean of the training examples) [2]
- TST - Template Selection Tracker initialised with the PSM points and updating the templates.

Results of applying these methods to the BIOID and XM2VTS data sets are shown in Figure 3.

Figures 3(a) and 3(b) show that the least successful method is simply using the average points from the global face detector with no local search (AVG dot-dash line). However the global face detector alone is reasonably successful finding 95% of facial feature points within 20% of the inter-ocular separation on the BIOID data set and 92% on the XM2VTS image database. Given the detected face region the feature localisation accuracy is improved on both



(a) BIOID Results



(b) XM2VTS Results

Figure 3. Cumulative distribution of point to point measure (m_e) on BIOID and XM2VTS data sets

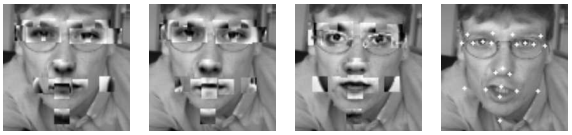
data sets by applying smaller feature detectors and using the PSM [7] constraint method (see PSM the dot-dash line). These feature points are used to initialise our local search methods.

Both Figure 3(a) and Figure 3(b) show similar trends. The best performing methods are the TST and AAM. The success rate of the TST (solid line) is greater than the AAM search (dashed line) at threshold values of $m_e < 0.075$. For $m_e > 0.075$ the AAM search is slightly more successful. On the BIOID test set the TST and AAM algorithms are able to locate approximately 90% of faces with a threshold accuracy of $m_e = 0.075$ and approximately 96% for $m_e = 0.150$. The dotted line in Figures 3(a) and 3(b) shows that using fixed templates (i.e. the mean of the training set templates with no updates) performs reasonably well, but always has a lower success rate compared to the full nearest neighbour update method (solid line).

Figure 4 shows an example of the TST search converging to a successful search solution on one example from the BIOID data set. The templates change at each iteration to resemble the search image.

[1] Note that the AAM formulation we use is actually the edge/corner AAM [12] method which we have found to be more effective than the basic texture method [4]

[2] Note this formulation is equivalent to the method described in [3]



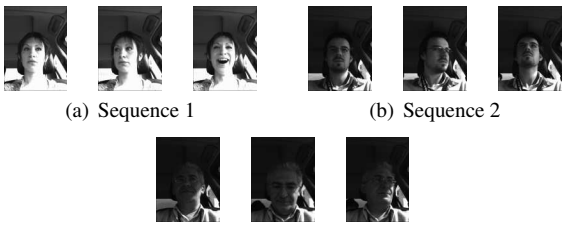
(a) Start Templates (b) After It1 (c) After It2 (d) Final Pts

Figure 4. Evolution of feature templates when searching a static image

4.3 Tracking Experiments

The TST algorithm automatically selects the feature templates to match the current image. Therefore the technique is well suited to tracking because the current templates and feature locations can be retained to search the next frame of the sequence. We test the Template Selection Tracker by applying it to three different sequences of the people driving in cars[3]. The test sequence involves a large amount of lighting variation and head movement and is thus a challenging data set. Each sequence contains approximately 1000 frames (taken at 10fps). See Figure 5 for example frames from the three test sequences.

The face rotates out of plane at some point in all three sequences. Therefore we use a quality of fit measure to test when the face has been lost and re-initialise by searching subsequent frames with the global face detector. The quality of fit measure used for the TST method is the shape constrained response score (see Equation 4). The AAM fit quality is the sum of residuals of the texture model fitted to the image.



(a) Sequence 1 (b) Sequence 2

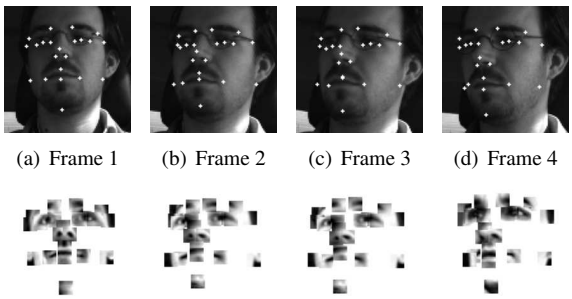
(c) Sequence 3

Figure 5. Example frames from car driver sequences

To provide ground truth for our experiments every 10^{th} frame (i.e once a second) is labelled by a human operator, provided all the facial features are visible. The distance measure for each labelled frame is Equation 5, unless the labelled face is undetected in the image, when the distance is recorded as infinite. The results of applying this detect/track scheme to the driver sequences are shown in Figure 6.

The graphs in Figure 6 show that the TST method (solid line) is the most effective tracking method over the three sequences, followed by the fixed template search (dotted line) and then the AAM tracker (dashed line). In Sequence 1 (see Figure 6(a)) the fixed template method gives very similar performance to the TST algorithm. This is probably due to the subject's facial texture in Sequence 1 lying close to the mean texture of the template appearance model.

Examples of the nearest neighbour templates selected by the TST when tracking the face in Sequence 2 are shown in Figure 7. Figures 7(a)-7(d) show frames where the head is rotated relative to the camera. Figures 7(e)-7(h) show that the nearest neighbour approach is able to select appropriate feature templates.



(a) Frame 1 (b) Frame 2 (c) Frame 3 (d) Frame 4

(e) Templates 1 (f) Templates 2 (g) Templates 3 (h) Templates 4

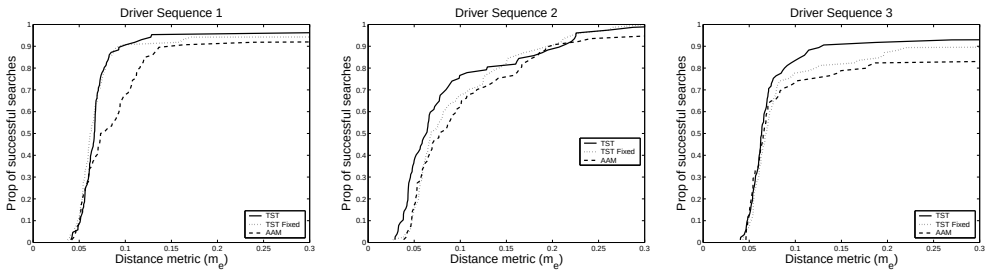
Figure 7. Selected templates and corresponding frames

4.4 Timings

When searching static images the global search followed by local feature detection requires ∼120ms. The time to apply the TST or AAM is less than ∼120ms, but in both cases depends on the number of iterations required to converge. The TST requires ∼40ms per search iteration. When searching static images two or three iterations are usually required. However when tracking, for most frames only one iteration is usually required. Therefore when searching a static image from the BIOID data set (384x286 pixels), with a P4 3Ghz processor the full search time is ∼240ms or 4 frames per second, but when tracking the TST search time drops to ∼40ms, or 25 frames per second. Note our face models are trained on many different people and can this match to almost anyone. The tracked face is not included in the training set.

Note the nearest neighbour matching time is less significant than the time required to compute template image responses and optimise the shape parameters. The technique

[3]None of the people were in the model training set



(a) Sequence 1 (b) Sequence 2 (c) Sequence 3

Figure 6. Cumulative distribution of point to point error measure

of first selecting the likely shapes then matching textures is sufficient for our case with 1052 training examples and 21 shape parameters for each face. However if the technique was applied to higher dimensional data it may be advisable to employ a more efficient nearest neighbour matching algorithm, for example see [9] for a review.

5 Summary and Conclusions

We have presented a novel algorithm to automatically find facial features in static images and video. The method fits a set of feature templates to the unseen image in an iterative manner. The search is constrained using a global shape model of the facial features, which allows for efficient matching. Also the feature templates are constrained using a combined shape and texture nearest neighbour method to select likely template textures from a training set.

Figure 3 shows that our approach outperforms a previous search method [3], which does not update the feature templates and gives similar performance to the AAM algorithm on static images. Figure 6 shows that our approach clearly outperforms the AAM algorithm when applied to a difficult set of in-car driving sequences.

Future work will involve extending our approach to model gradients rather than normalised pixel values, as this has been shown to improve the AAM search [12]. We may also investigate automatic model building methods, as presently the set of features and template region sizes are picked by hand, which may well be sub-optimal. Additionally the simple nearest neighbour texture model and shape constraint search method may easily be extended to 3D for use in high dimensional medical data.

In conclusion the Template Selection Tracker method is a simple and efficient tracking algorithm. Our shape/texture nearest neighbour matching technique is able to reliably select appropriate templates that can then be used in a robust shape constrained search. We demonstrate that the new TST approach outperforms the AAM when used to track human faces.

References

[1] 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.

[2] T. F. Cootes and C. J.Taylor. Active shape models. In David Hogg and Roger Boyle, editors, 3^{rd} *British Machine Vision Conference*, pages 266–275. Springer-Verlag, September 1992.

[3] D. Cristinacce and T. Cootes. A comparison of shape constrained facial feature detectors. In 6^{th} *International Conference on Automatic Face and Gesture Recognition 2004, Seoul, Korea*, pages 375–380, 2004.

[4] D. Cristinacce, T. Cootes, and I. Scott. A multi-stage approach to facial feature detection. In 15^{th} *British Machine Vision Conference, London, England*, pages 277–286, 2004.

[5] N. Dowson and R. Bowden. Simultaneus modeling and tracking (smat) of feature sets. In *Computer Vision and Pattern Recognition Conference 2005, San Diego, USA*, 2005.

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

[7] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61, 2005.

[8] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz. Robust face detection using the hausdorff distance. In 3^{rd} *International Conference on Audio- and Video-Based Biometric Person Authentication 2001*, 2001.

[9] J. McNames. A fast nearest-neighbor algorithm based on a principal axis search tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):964–976, 2001.

[10] 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.

[11] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[12] I. M. Scott, T. F. Cootes, and C. J. Taylor. Improving appearance model matching using local image structure. In *Information Processing in Medical Imaging,* 18^{th} *International Conference*, pages 258–269, July 2003.

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