

# Finding Image Features Using Deformable Templates And Detailed Prior Statistical Knowledge

Alan Bennett\* and Ian Craw

Department of Mathematical Sciences  
University of Aberdeen  
Aberdeen AB9 2TY, Scotland

## Abstract

Much work in image processing has been devoted to generating filters to detect low level image features, e.g. edges, peaks, valleys. Objects are then located or recognised in the image by using the output from these filters.

We present methods which can interpret the output from filters in order to locate known objects in images. The algorithms use statistical knowledge about the variation of the shape of objects being searched for in order to guide the search to only feasible regions.

The main techniques used are various deformable template algorithms where optimisations are achieved by random sampling and simulated annealing to avoid non-global extrema. The particular application here is for locating facial features including head outlines, where the results give key locations on the face and allow approximate geometric representations of the features to be reconstructed.

## 1 Introduction

Much early work in image processing was devoted to generating filters to detect low level image features, e.g. edges, peaks, valleys. Objects are then located or recognised in the image by using the output from these filters.

The following work is an investigation into methods of interpreting the output from simple feature detectors in order to recognise and locate objects in grey scale images. We present a method for locating image feature outlines using probabilistic methods first introduced by Grenander et al [1]. We have extended these techniques to deal with non-binary images. In addition, we use detailed statistical knowledge to search for internal features by using more sophisticated geometric models such as those employed by Yuille et al [2]. We

---

\*Supported by a SERC studentship.

have implemented various algorithms which attempt to locate these objects irrespective of scale and orientation, and in the presence of noise, background textures and occlusions.

We explain how we build models of particular features, e.g. eyes, from a parameterised geometric template in conjunction with some measure of how well the model is in agreement with the image. We conclude by giving examples of the algorithms applied to both hands and faces where we wish to locate the outline and, for the faces, the internal features as well.

The work demonstrates the advantages achieved by using the statistical information available to reduce the combinatorially large search space by guiding the search to only feasible representations.

## 2 Building Feature Models

Associated with each object we locate, we require a geometric representation of the object, which can be expressed by a small number of parameters, and a measure of how well the representation matches the image.

The geometric model is composed of several simple geometric shapes, e.g. polygons, circles, parabolae. One such model is the eye template used by Yuille [2] where an eye is modelled by two parabolae, representing the eye lids, and a circle, representing the iris.

The model may be built ad-hoc, although a template generated using a sample of measured objects is likely to be more reliable and is thus preferred.

### 2.1 The Fitness Function

To locate features, we must not only find them, but we must know when we have found them. We, therefore, require methods for combining the filtered image information to generate functions relating how well the templates match the image, and how well each shape matches the typical shape. This *fitness function* is calculated in two distinct parts.

The first is a measure of the deviation from the initial template shape (the *shape score*). This *shape score* is itself calculated in two parts: the prior belief about the distribution of the template parameters (calculated as a sum of the deviations of each parameter), and how closely the template resembles the required shape, independent of scale and orientation, (calculated in terms of the relationship between related parameters).

The second part of the fitness function is a measure of how well the feature being modelled is in accordance with the grey scale image. This expresses how well the salient image features are being modelled by the geometric shapes of the template, (the *image score*). For example, in calculating the fitness of an eye, we maximise the mean valley intensity within the iris, maximise the mean edge score around the eyelids and the iris, and maximise the peak intensity in the whites of the eye. The peak and valley intensities are found by using morphological filters as described by Serra [3].

## 2.2 The Confidence Function

In addition to the fitness value described earlier, we require a *confidence value*. We wish this *confidence value* to reflect how confident we are that the algorithm has found an extremum as required and also that that extremum corresponds to the particular feature we seek.

A naive confidence value may involve comparing the edge score with the maximum edge gradient and similarly for any other filtered output. Unfortunately, this is merely an indication of how well the iterate performs relative to all possible iterates.

A better method is to look for peaks in the fitness score. A feature which has been found successfully scores well, but neighbouring locations should score less well. The rate of change of fitness score near a peak would then indicate how well the model fits the feature and hence provides a more reliable confidence value in the location.

## 3 Prior Statistical Knowledge

The algorithms use detailed statistical information about the nature of the variation of the shape of each feature. This statistical information is generated by estimating the distributions of the model parameters given a large example set of objects.

Clearly, the parameters for each feature are not independent, hence we express the parameters in terms of related parameters, e.g. the height of an eye as a function of its width.

The particular prior knowledge we use consists of the known locations of 37 key points in each of 1000 faces. This enables very reliable estimations of the required distributions and provides a means for generating templates.

## 4 The Algorithms

In this section, we describe the algorithms used to locate facial features. The simplest algorithm finds the head outline irrespective of the internal features. This outline is used by the next algorithm to guide the search for the eyes, which in turn guides the search for the mouth. The third algorithm is more general, finding the outline and the internal features together. In this way, the success of any one feature finder can influence the search for any of the other features.

### 4.1 The Outline Finder

The head outline is found using techniques similar to those employed by Grenander et al [1, 4] and [5] where a polygonal template outline is transformed at random to fit the data. The template is generated by averaging the 10 locations of each of the 1000 measured faces corresponding to the head outline. Some extra points are added by hand, to make the shape less jagged and more head-like.

The approximate location, scale and orientation of the head is found by repeatedly deforming the whole template at random by scaling, rotation and

translation, until it matches best with the image whilst remaining a feasible head shape. This feasibility is determined by imposing statistical constraints on the range of allowable transformations. A further refinement is then achieved by transforming the individual vectors within the polygon under certain statistical constraints. The optimisation, in both stages, is achieved by using simulated annealing.

## 4.2 The Sequential Algorithm

Here we use the result of the outline finder to guide the search for the internal features. The locations of the eyes with respect to the outline, are known statistically, and hence can be sampled at random. Similarly, we sample the mouth with respect to the eyes and the outline.

We sample the locations in two distinct ways. The first samples the locations and sizes directly from the prior distributions. This method allows a large variety of features to be sampled but is unlikely to find the correct solution due to the large number of parameters.

The second method samples locally by varying the parameters, about an existing good estimate. This allows the algorithm to home in on correct solutions from close estimates.

A combination of these two strategies is used to perform the required sampling.

## 4.3 A More General Approach

A more sophisticated approach samples features concurrently and then accepts or rejects the whole iterate. It is important here to emphasize that the head outline is a feature. The algorithm samples an outline and then the internal features accepting the outline only if the internal features match the image also, *or* samples internal features and then an outline with respect to the internal features.

## 5 Results

We first demonstrate the outline finder locating a hand in both a clean and a noisy image, signal to noise ratio of 1:1, (see figures 1 and 2), showing the inherent tolerance of noise due to the averaging in the edge score and the strength of the prior knowledge about the shape.

The algorithms have also been tested on over 100 mug shot style head and shoulder images, at a resolution of  $128 \times 128$  pixels, and also a few images with background texture. Here, the outline finder has proven very successful for the majority of test images, (see figure 3). The algorithm was successful in locating the head in more than 90% of the images, missing the head completely only once. Indeed successful approximations have been achieved for every test image to date, with only minor adjustments to the parameters. Where the algorithm does not return the desired outline, the failure is usually caused by relatively weak edge information around the chin or the top of a bald or grey head on a light background.

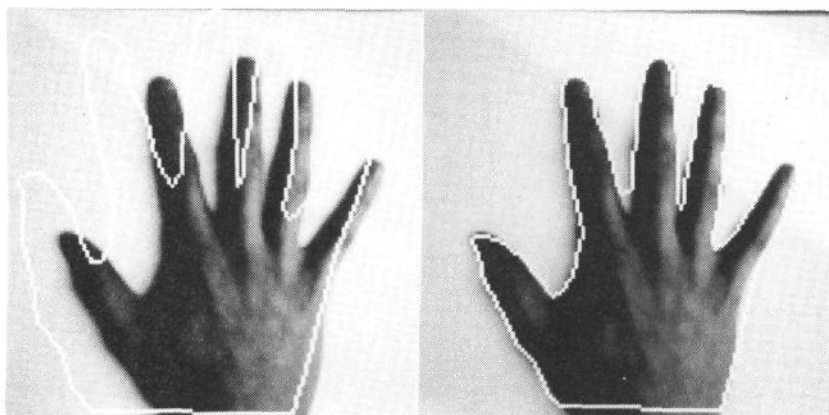


Figure 1: Finding a hand outline in a clean image. The left image shows the original template position, and the right image shows the resulting outline.

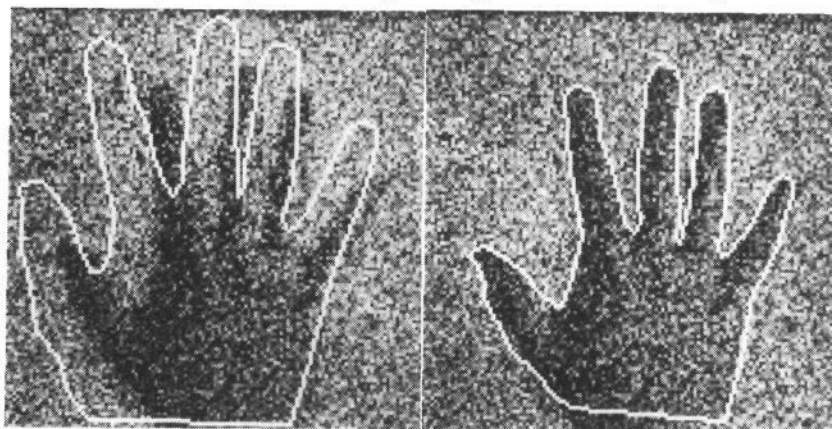


Figure 2: Finding a hand outline in the presence of noise. As for figure 1 but with added gaussian noise with signal to noise ratio of 1:1.

Results of the outline finder should be contrasted with those of Waite and Welsh [6], where the outline is found using snakes. We claim that the inclusion of prior knowledge about expected head shapes enables us to obtain superior results especially in the presence of noise or background information.

Given a fair approximation to the outline, the locations of the internal features can be estimated to within a few pixels. From this, already close, start, the feature locations are almost always found correctly, (see figure 4).

The reliability of the outline and eye finders has enabled them to be included in an expert system [7] which attempts to locate 30 key points in face images with a success rate of 80-90%.

Although still in the early stages, (in particular, very little work has been devoted to the mouth detector), the general algorithm has produced promising results (see figure 5), where the location of the internal features support the outline and draw it towards a more plausible head outline. One worry about the previous algorithms is that they make inferences about the shape of the head by



Figure 3: Finding head outlines.



Figure 4: Finding internal features from a head outline.

matching with the head outline. Unfortunately, the shape of the head outline depends highly on the hairstyle which can be hugely variable. By matching with the internal features, we can hope to generate a more reliable face shape for all such variable images.

## 6 Conclusion

We claim that we have demonstrated novel and effective deformable template algorithms with considerable success in locating image features. We have also shown the advantages of using detailed statistical knowledge to guide the search to avoid a combinatorial explosion when searching for many features described by a large parameter set.



Figure 5: Concurrent search for features including outline.

## References

- [1] U. Grenander, Y. Chow, and D.M. Keenan. *HANDS. A Pattern Theoretic Study of Biological Shapes*. Springer-Verlag, New York, 1991.
- [2] A. Yuille, D. Cohen, and P. Hallinan. Facial Feature Extraction by Deformable Templates. Technical Report CICS-P-124, Center for Intelligent Control Systems, Cambridge, MA 02139, March 1988.
- [3] Jean Serra. *Image Analysis and Mathematical Morphology*, volume 1. Academic Press, 1988. pp424-478.
- [4] U. Grenander and D.M. Keenan. Towards automated image understanding. *Journal of Applied Statistics*, 16(2):207-221, 1989.
- [5] A. Knoerr. Global models of natural boundaries: Theory and applications. Pattern Analysis Technical Report 148, Brown University, Providence, RI, 1988.
- [6] J.B. Waite and W.J. Welsh. An Application of Active Contour Models to Head Boundary Location. In *Proceedings of the British Machine Vision Conference*, pages 407-412, 1990.
- [7] D.I. Tock, I.G. Craw, and J.R. Lishman. A Knowledge Based System for Measuring Faces. In *Proceedings of the British Machine Vision Conference*, pages 401-406, 1990.
- [8] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE: Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721-741, November 1984.
- [9] D.W. Murray, A. Kashko, and H. Buxton. A parallel approach to the picture restoration algorithm of Geman and Geman on an SIMD machine. *Image and Vision Computing*, 4(3):133-142, August 1986.