Locating Salient Object Features

K.N.Walker, T.F.Cootes and C.J.Taylor
Dept. Medical Biophysics,
Manchester University, UK
knw@sv1.smb.man.ac.uk

Abstract

We present a method for locating salient object features. Salient features are those which have a low probability of being mis-classified with any other feature, and are therefore more easily found in a similar image containing an example of the object. The local image structure can be described by vectors extracted using a standard 'feature extractor' at a range of scales. We train statistical models for each feature, using vectors taken from a number of training examples. The feature models can then be used to find the probability of misclassifying a feature with all other features. Low probabilities indicate a salient feature. Results are presented showing that salient features can be relocated more reliably than features chosen using previous methods, including hand picked features.

1 Introduction

When analysing images of a class of object, we frequently begin by locating 'salient' features to facilitate further processing. Such features are often selected by the system designer. In this paper we describe how salient features can be chosen automatically.

Our previous approach [9][10] used a feature extractor to construct feature vectors over a range of scales, for each image point within a single object example. In order to locate salient points we estimated the probability density function of feature space, and selected features which lay in low density areas of the space.

In this paper we develop an approach which can take advantage of a set of training images, assuming a correspondence between them is known. A statistical model is constructed for each possible feature, representing the probability density function (p.d.f.) for the corresponding feature vectors. By comparing the p.d.f. of any feature with those of all others, we estimate the probability of misclassification. Salient features are those with a low misclassification rate.

In the following we give an overview of the original method, describe the new approach in detail and present results of an experiment comparing the two approaches.

2 Background

Objects typically have a large number of features. Many people have used subsets of features to accurately locate objects. It is common practice for the system designer to select this subset of features manually, a task which is subjective but critical to the success

of the system. Face recognition is one area in which manually selecting features with which to interpret a face is common. Tao *et al* [7] selected 39 facial features which they tracked using a probabilistic network. Lyons *et al* [5] used Gabor Wavelets to code 34 hand selected facial features in order to identify facial expressions. Also, Cootes *et al* [2] builds models of object shape by placing hand chosen landmarks on a number of training examples.

We believe that there is an optimum subset of these features which best determine the object, and attempting to select them manually risks compromising system performance.

Many authors have shown that using the saliency of image features can improve the robustness in object recognition algorithms [1] [8] [6], but this typically been applied to finding salient segments on an object boundary.

3 Method 1: Selecting Salient Features from a Single Training Example

The aim is to locate salient features, those which are most likely to be relocated correctly in a subsequent image. Given only one example of the object, the best we can do is to attempt to find those features which are significantly different to all other features in the object. Ideally, these features would occur exactly once in each example of the object.

For every pixel in the image we construct several feature vectors, each of which describes a feature centred on the pixel at a range of scales. The full set of vectors describing all object features, forms a multi-variate distribution in a *feature space*. By modelling the density in feature space we can estimate how likely a given feature is to be confused with other features. Salient features lie in low density areas of feature space.

In the rest of this section we describe how we modelled the density of feature space and how this can be used to select salient features. See [9][10] for a more detailed description.

3.1 Modelling the Density of Feature Space

We estimate the local density $\hat{p}$ at point $\mathbf{x}$ in a feature space by summing the contribution from a mixture of m Gaussians:

$$\hat{p}(\mathbf{x}) = \sum_{i=1}^{m} w_i G(\mathbf{x} - \mathbf{m_i}; \mathbf{\Sigma_i}) \tag{1}$$

where $G(\mathbf{x}; \mathbf{\Sigma})$ gives the probability of $\mathbf{x}$ being part of the normalised multi-variant Gaussian distribution, with covariance $\mathbf{\Sigma}$ and a mean of 0.

We use two methods for choosing the parameters.

- *Kernel method*: The Kernel method positions Gaussians at all the samples in the distribution. In this case the parameters are $m = N$,$w_i = \frac{1}{N}$, $\mathbf{m_i} = \mathbf{x_i}$ and $\mathbf{\Sigma_i} = hS$, where N is the number of samples in the distribution, S is the covariance of the whole distribution and h is a scaling factor. This method is of order n^2.

- *Sub-sampled Kernel method*: This method attempts to approximate the Kernel method by placing gaussian kernels at a randomly selected n_s of the original n points. Evaluation of the probability density function is order n_s^2, so can be much more efficient than the kernel method, but this comes with some loss of accuracy.

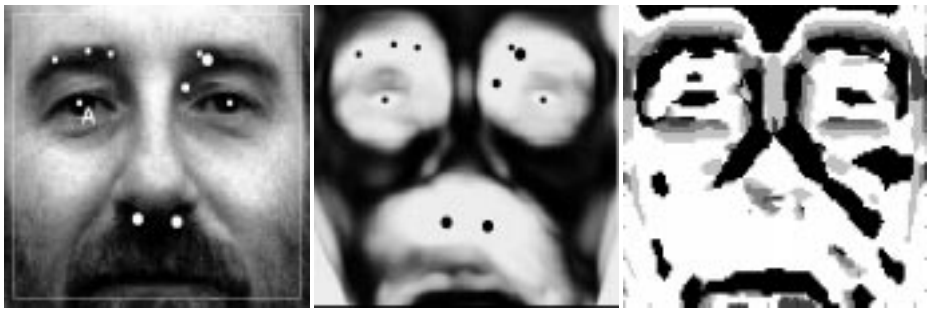
We found that the Kernel method gave the best results but was to computationally expensive if the features were extracted from a number of scales. The Sub-sampling method was found to give a good approximation to the Kernel method.

3.2 Selecting Salient Features

The density estimate for each feature vector $\mathbf{v}(\sigma)$ corresponds directly to the saliency of the feature at scale σ. The lower the density, the more salient the point.

A saliency image can be constructed by setting each pixel to the density of the corresponding feature vector at the most salient scale for that pixel. The most salient features are then found by locating the lowest troughs in the saliency image. A scale image can be constructed by plotting the most salient scale at each pixel. This shows the scale at which each region of the image is most salient.

Figure 1(b) is the saliency image obtained from the image in Figure 1(a) using the Sub-sampled Kernel method with 50 Gaussians. The bright regions are the most salient. Figure 1(c) is the corresponding scale image. The peaks of the saliency image are superimposed on the original image. The size of the points indicate the scale at which the features are salient.



(a) (b) (c)

Figure 1: (b) is the saliency image obtained from image (a), and (c) is the corresponding scale image.

4 Method 2: Selecting Salient Features From Many Training Examples

We have shown that locating salient features using one training example results in features which can be found in unseen object examples more reliably than hand chosen features [10]. When training on only a single example, we define a salient feature to be one

which is significantly different to all others in that example. However, this does not take account of how *reliable* the feature is, whether it occurs in all examples of the object or whether it varies enough to be confused with other object features. These factors can not be determined by considering just a single example.

We present a new approach which attempts to model how each individual feature varies over training examples. This is done by extracting a feature vector describing a particular feature from all training examples. The set of feature vectors which describe the feature is then used to create a feature model. The saliency of the feature is determined by using the feature models to calculate the probability of mis-classifying the feature with any other features from within the object.

In the following we define how the feature models are built and how the saliency measures are calculated from the feature models.

4.1 Building the Feature Models

In order to extract a feature vector representing the same feature in all training examples, we first establish a correspondence between all training examples. This is currently done by interpolating between a set of common landmarks placed on all training examples. Figure 2 shows some examples of such landmarks placed on faces.

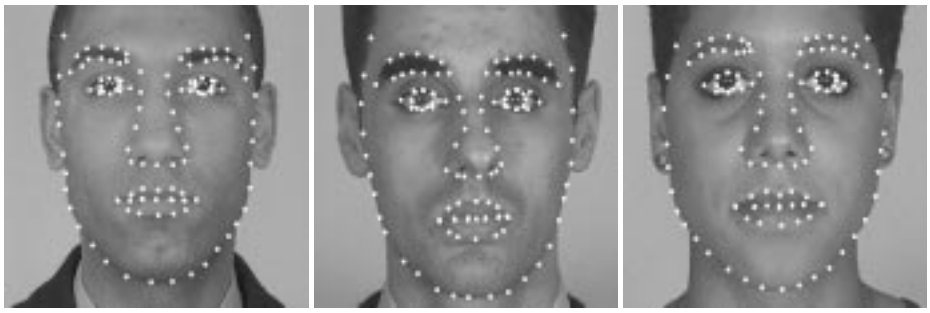


Figure 2: Examples of face images with common landmarks.

We then define the features we wish to model at each scale. We calculate mean face [3] based on the training examples. We then model one feature for each pixel in the mean face. The approximate number of pixels in the mean face, n_f, can be set according to the computational power available. n_f defines the number of features per scale. The total number of features modelled is $n_s.n_f$, where n_s is the number of scales.

The p.d.f. for feature i, f_i, is then modelled using a multi-variate gaussian with mean μ_i and covariance Σ_i (determined from the training set).

4.2 Calculating a Features Saliency

Given a probability density function, f_i, for each feature it is now possible to calculate a features saliency. The saliency of feature i, s_i, is given by the probability of not mis-classifying feature i with any other object feature. Thus:

$$s_i = 1 - \frac{1}{n_f - 1} \sum_{j=1; j \neq i}^{n_s . n_f} \epsilon(f_i, f_j) \tag{2}$$

where $\epsilon(f_i, f_j)$ is the probability of misclassifying feature f_i with feature f_j.

4.2.1 Calculating $\epsilon(f_i, f_j)$

Consider first the 1D case. We wish to calculate $\epsilon(f_i, f_j)$, the probability of misclassifying a sample from distribution f_i as being from f_j. In the case where f_i and f_j are gaussians, an analytic solution using error functions exists (see Appendix A).

Typically the dimensionality of the feature vectors is much higher than 1. With higher dimensions the misclassification regions become increasingly complicated volumes, making $\epsilon(f_i, f_j)$ hard to calculate. Also, to calculate the saliency measure for all features, $\epsilon(f_i, f_j)$ must be evaluated approximately $\frac{1}{2}(n_s . n_f)^2$ times, where n_s is the number of scales analysed. Because of the complexity of calculating $\epsilon(f_i, f_j)$ in high dimensional spaces and the large number of times it must be evaluated it is necessary to approximate $\epsilon(f_i, f_j)$.

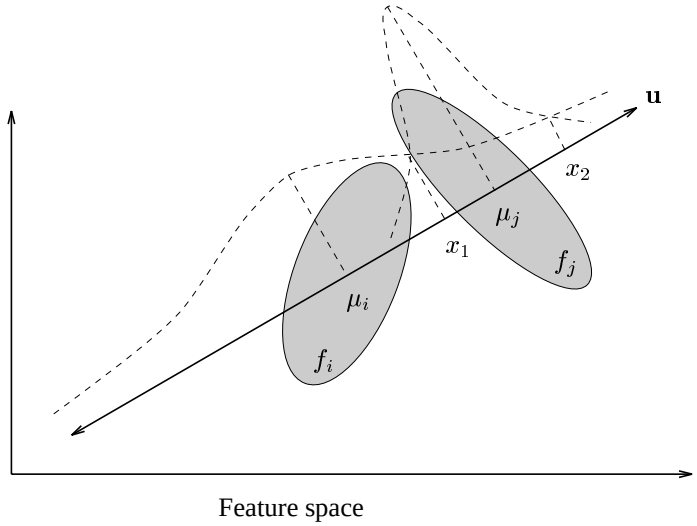


Figure 3: Illustration of how a multi-variate gaussian classification can be simplified to a single dimensional problem.

We approximate $\epsilon(f_i, f_j)$ by simplifying the problem to a single dimension, where it can be solved using error functions. This process is illustrated in Figure 3. The first step is to construct a new one dimensional space. This is simply the axis which passes through the mean of both features distributions f_i and f_j. This axis is labelled $\mathbf{u}$ in Figure 3.

The variance, σ_i, along $\mathbf{u}$ due to distribution f_i is given by:

$$\sigma_i{}^2 = \mathbf{u}^T . \Sigma_i . \mathbf{u} \tag{3}$$

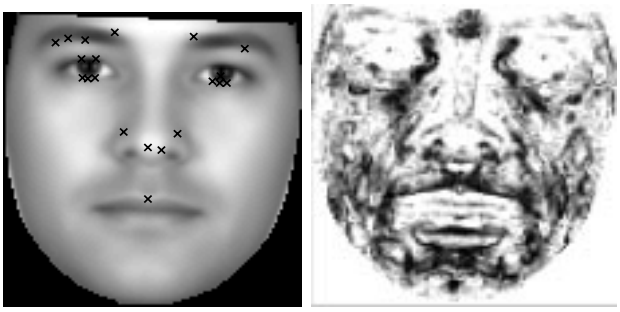
where Σ_i is the covariance matrix of feature model f_i.

μ_i, μ_j, σ_i and σ_j now form a one dimensional problem which can be solved using error functions as shown in Appendix A.

4.3 Constructing a Saliency Image

Once the Saliency measure, s_i, of each feature has been found the result can be visualised by constructing a saliency image. This is done by taking the mean image of the object and at each pixel plotting the saliency measure corresponding to the most salient feature centred on that pixel. The resulting image indicates which areas of the object are the most salient.

Figure 4(b) shows an example of such a saliency image which was trained on approximately two hundred images of faces. Figure 4(a) shows the mean face with the peaks of the saliency image super-imposed.



(a) (b)

Figure 4: (b) is the saliency image obtained from approximately 100 face images, white regions are most salient. (a) show the peaks of the saliency image superimposed onto a 'mean' face

5 Results

In order to quantify if the features selected using method 2 are more salient than those of method 1, we addressed the question 'how successfully can we find the features in unseen images?'. We located the 20 most salient features according to method 1 (Figure 5(a)) and according to method 2 (Figure 5(b)). We then attempted to locate these features in 188 unseen images of faces. We recorded the rank of the correct match for each search.

To provide a means of contrast we also selected 20 features by hand (Figure 5(c)) and 20 randomly selected features (Figure 5(d)), and repeated the test on these.

The result of tests is shown in Figure 5. The graph shows the percentage of successful searches according to the number of false positives. It can be seen that, as expected, the randomly selected features do worse. The hand selected features approximately double the performance of the random selected features. Features selected by method 1 do better that hand selected points, as previously reported [10], but the features which were most successfully found in the unseen images were those selected using method 2.

All the results used Cartesian Differential invariants [4] as the feature extractor.

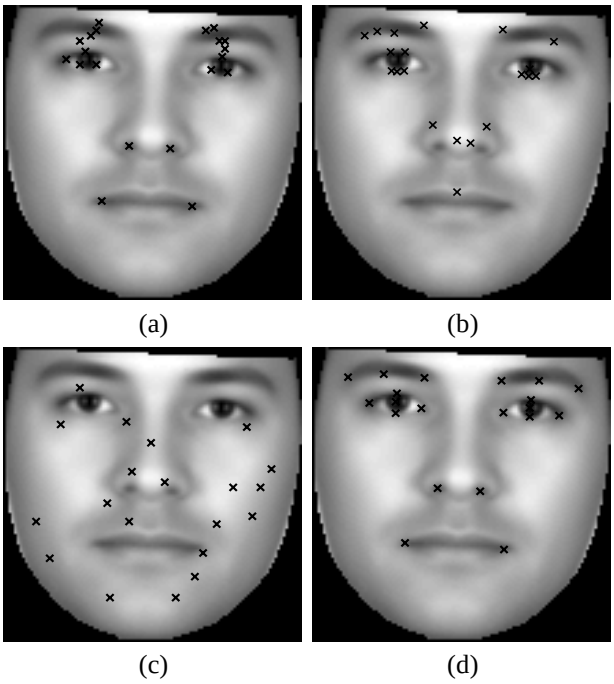


Figure 5: The most salient features according to (a) method 1, (b) method 2, (c) randomly selected and (d) hand selected. These are the features used in the results.

6 Discussion

We have described how a probabilistic measure of saliency, calculated from a number of object examples, can be used to select those object features least likely to generate false positive matches.

We have also shown that the salient features selected can be found with a greater degree of success in unseen object examples that features selected using previous methods [10], including hand selected features.

We have applied the notion of saliency to improve the robustness of one object interpretation task, locating new instances of an object. The robustness of other interpretation tasks can also benefit from the application of saliency, for example, classifying face gestures. Saliency could be applied to locate the facial features which most discriminate between gestures.

We anticipate that the method of detecting salient points and of locating their positions in new images will prove useful both for generating cues to prime models for search, and to help to automatically train statistical appearance models by locating common landmarks on a set of training images.

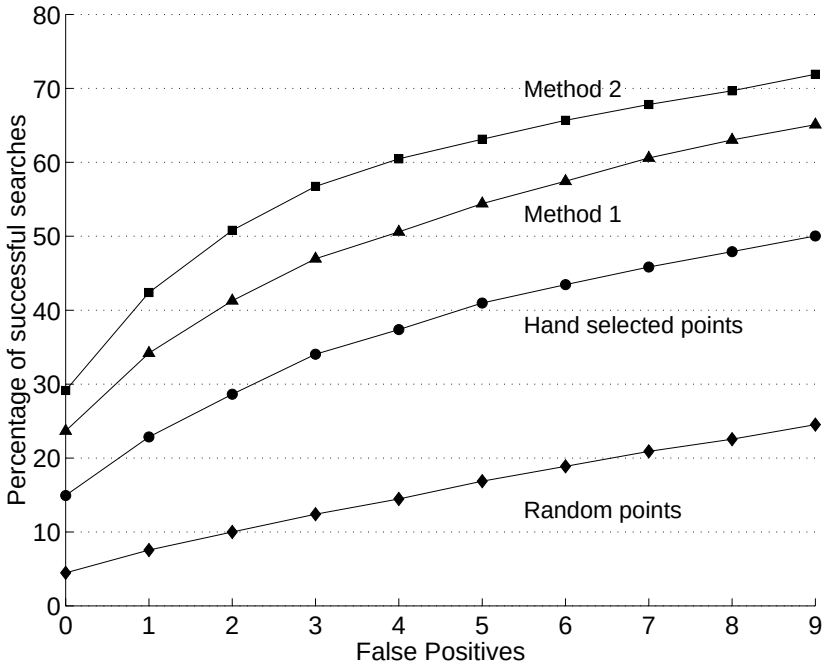


Figure 6: Graph illustrating the percentage of successful searches according to the number of false positives.

Appendix A: Calculating $\epsilon(f_i, f_j)$ in the 1D case

In the one dimensional space, the probability of a sample x belonging to feature i is given by:

$$p_i(x) = \frac{1}{\sqrt{2\pi {\sigma_i}^2}} e^{-\frac{(x-\mu_i)^2}{2{\sigma_i}^2}} \tag{4}$$

where σ_i and μ_i are determined from the training set.

Assuming all features are equally likely the Likelihood ratio test suggests classifying x as class i if $p_i(x) > p_j(x)$. This splits the space up into two or three regions, where the boundary points, x, satisfy:

$$p_i(x) = p_j(x) \tag{5}$$

Figure 7 illustrates these regions. There are three situations depending on the values of σ_i and σ_j. Taking logs from equation 5 results in a quadratic with roots equal to x_1 and x_2. Solving the quadratic gives:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{6}$$

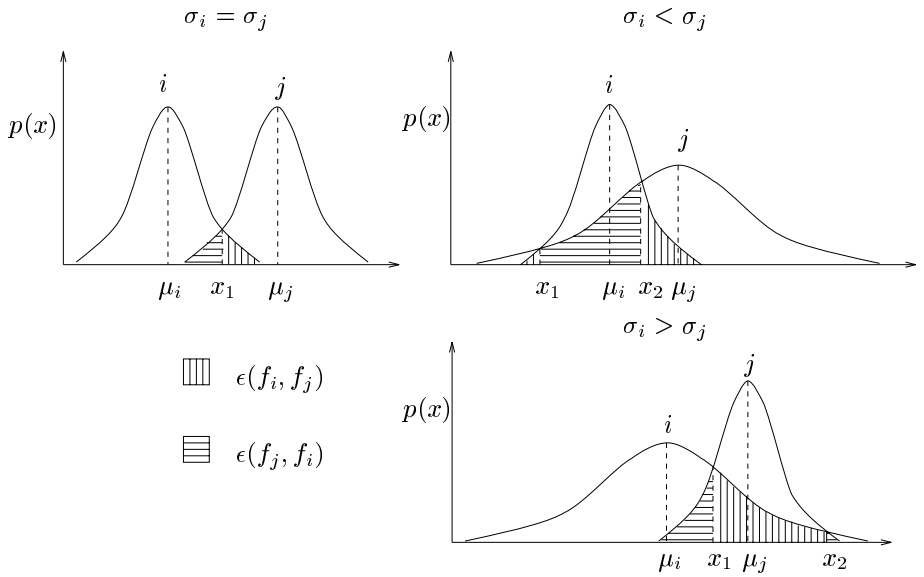


Figure 7: Illustrates the areas which need to be calculated in order to evaluate $\epsilon(f_i, f_j)$ in the 1D case. There are three cases depending on the values of σ_i and σ_j.

where

$$a = \frac{1}{{\sigma_j}^2} - \frac{1}{{\sigma_i}^2} \tag{7}$$

$$b = 2(\frac{{\mu_i}^2}{\sigma i^2} - \frac{{\mu_j}^2}{\sigma j^2}) \tag{8}$$

$$c = \frac{{\mu_j}^2}{{\sigma_j}^2} - \frac{{\mu_i}^2}{{\sigma_i}^2} - 2ln(\frac{\sigma_i}{\sigma_j}) \tag{9}$$

x_1 and x_2 can be simplified further by making $\mu_i = 0$.

$\epsilon(f_i, f_j)$ can now be approximated in one of the three following ways, depending on the value of σ_i and σ_j:

if $\sigma_i = \sigma_j$ then

$$\epsilon(f_i, f_j) = \frac{1}{2}(1 - erf(\frac{x_1 - \mu_i}{\sqrt{2}\sigma_i})) \tag{10}$$

if $\sigma_i < \sigma_j$ then

$$\epsilon(f_i, f_j) = 1 + \frac{1}{2}(erf(\frac{x_1 - \mu_i}{\sqrt{2}\sigma_i}) - erf(\frac{x_2 - \mu_i}{\sqrt{2}\sigma_i})) \tag{11}$$

if $\sigma_i > \sigma_j$ then

$$\epsilon(f_i, f_j) = \frac{1}{2}(erf(\frac{x_2 - \mu_i}{\sqrt{2}\sigma_i}) - erf(\frac{x_1 - \mu_i}{\sqrt{2}\sigma_i})) \quad (12)$$

Thus, equations 10, 11 and 12 can then be substituted into equation 2 to obtain a measure of feature saliency.

References

[1] R. C. Bolles and R. A. Cain. Recognising and locating partially visible objects: the local-feature-focus method. *Int. J. Robotics Res*, 1:57–82, 1982.

[2] T.F. Cootes and C.J. Taylor. Locating objects of varying shape using statistical feature detectors. *Computer Vision - ECCV*, II:464–474, 1996.

[3] G.J. Edwards, T.F. Cootes, and C.J. Taylor. Interpreting Face Images using Active Appearance Models. In *International Workshop on Automatic Face and Gesture Recognition 1998*, pages 300–305, Nara, Japan, 1998.

[4] Luc M J Florack, Bart M ter Haar Romeny, Jan J Koenderink, and Max A Viergever. Scale and the differential structure of images. *Image and Vision Computing*, 10(6):376–388, July/August 1992.

[5] Micheal Lyons and Shigeru Akamatsu. Coding Facial Expressions with Gabor Wavelets. In *International Workshop on Automatic Face and Gesture Recognition 1998*, pages 200–205, Nara, Japan, 1998.

[6] T. N. Mudge, J. L. Turney, and R. A. Voltz. Automatic generation of salient features for the recognition of partially occluded parts. *Robotica*, 5:117–127, 1987.

[7] Hai Tao, Ricardo Lopez, and Thomas Huang. Tracking Facial Features Using Probabilistic Network. In *International Workshop on Automatic Face and Gesture Recognition 1998*, pages 166–170, Nara, Japan, 1998.

[8] J. L. Turney, T. N. Mudge, and R. A. Voltz. Recognising partially occluded parts. *IEEE Trans. PAMI*, 7:410–421, 1985.

[9] K.N. Walker, T.F. Cootes, and C.J. Taylor. Correspondence Using Distinct Points Based on Image Invariants. In *British Machine Vision Conference 1997*, pages 540–549, 1997.

[10] K.N. Walker, T.F. Cootes, and C.J. Taylor. Locating Salient Facial Features Using Image Invariants. In *International Workshop on Automatic Face and Gesture Recognition 1998*, pages 242–247, Nara, Japan, 1998.