Statistical Grey-Level Models for Object Location and Identification

T.F. Cootes, G.J. Page, C.B. Jackson, C.J.Taylor

Department of Medical Biophysics
University of Manchester
Oxford Road
Manchester M13 9PT
email: bim@sv1.smb.man.ac.uk

Abstract

This paper presents a new method for modelling and locating objects in images for applications such as Printed Circuit Board (PCB) inspection. Objects of interest are assumed to exhibit little variation in size or shape from one example to the next, but may vary considerably in grey-level appearance. Simple correlation based approaches perform poorly on such examples. We demonstrate how a statistical model based approach combined with a multi-resolution search can accurately locate objects and reliably distinguish between good and bad components. We describe a 'bootstrap' approach to training and a method of automatically refining the final model to improve its performance. We demonstrate the method on PCB inspection, showing the approach is robust enough for use in a real production environment.

Introduction

Many industrial inspection problems consist of locating and classifying rigid components, for example electronic devices on a printed circuit board. Typically the size and shape of each component is defined within strict tolerances, but the surface appearance (colour, texture, surface markings etc) can vary, sometimes dramatically. These variations, combined with lighting changes and camera noise, can easily confound naive correlation based methods for locating the components. To locate and classify a component accurately we need a good model of its typical appearance and how the appearance can vary. In this paper we describe how statistical models based on the variation in grey-levels across the component can be used to achieve this end. We compare different statistical models and model building methods. By combining several techniques we can discriminate between true and false examples with considerable confidence. By testing on data sets containing large numbers of examples we demonstrate that our techniques allow us to predict the probability distribution of quality-of-fit parameters for a given component. We can thus estimate a threshold which will fail no more than a given number of correct components (eg 1 in 1000) – an important factor in setting up an inspection system.

In the remainder of the paper we review related work, describe the modelling and search techniques and give results of experiments to test the system.

BMVC 1995 doi:10.5244/C.9.53

Background

The simplest approach to locating a rigid component in an image is to use a correlation based matching technique using a single example of the object as a template [1]. This method will fail when the appearance of the component varies or there is too much noise.

A more robust approach is to build a statistical model of the grey–levels in the region of the component. Turk and Pentland [2] use Principle Component Analysis (PCA) to build 'eigen–feature' models of the grey–levels across a face and face features, which they then use in a correlation–like framework to detect faces and features in new images. These models can cope with considerable variation and usually give a distinct peak in response at the target location. This work was extended in a probabalistic framework by Moghaddam and Pentland [3], who consider both uni–modal and multi–modal gaussian distribution models. A similar approach was used by Cootes *et al* to model the grey–levels in regions around the model points of their Active Shape Models [4]. They went on to demonstrate a multi–resolution search technique in which grey–level models were trained and used at each level of a gaussian image pyramid [5].

We have found that although the eigen–feature models combined with the multi–resolution search technique produce very promising results they occasionally fail unexpectedly. For instance, models trained to locate a surface mount resistor can occasionally (about 1% of the time) fit better to an empty section of printed circuit board than to a true component. This failure rate makes them impractical for most industrial inspection problems.

Below we describe a more detailed statistical model which, when combined with more accurate location methods, leads to a technique which is considerably more robust.

Method

In this section we describe the model building and search techniques that we have developed. Modelling involves locating typical examples to create a training set and building a statistical model of the variation of grey–levels across the examples.

The Basic Model

We wish to locate accurately examples of a given component in a new image, typically in order to decide a) if the correct component is present in a given region and b) if the component is mounted correctly to within some tolerance on its position or orientation. We wish to be able consistently to distinguish the component both from the background and from other, possibly quite similar, components.

To deal with variations in appearance of components we will use statistical models derived from a set of training examples. In the experiments described below these are simply rectangular regions of image containing instances of the component of interest (see Figure 3). For each n_x x n_y patch we sample the image at pixel intervals to obtain a $n = n_x n_y$ element vector, $\mathbf{x}$. In general any shape of region is acceptable, as long as a consistent sampling is defined for it.

Given a set of s such example vectors, $\mathbf{x}_i$ $(i = 1..s)$, a linear 'eigen'-model can be built by applying a Principle Component Analysis to obtain the mean, $\bar{\mathbf{x}}$, and t principle modes of variation represented by the n x t matrix of eigenvectors, $\mathbf{P}$. The value of t is chosen so that the model represents a suitable proportion of the variation in the training set (eg 95%) [4]. A particular patch $\mathbf{x}$ can be approximated by

$$\mathbf{x} \approx \mathbf{x}' = \bar{\mathbf{x}} + \mathbf{Pb} \qquad \text{where} \quad \mathbf{b} = \mathbf{P}^T(\mathbf{x} - \bar{\mathbf{x}})$$

We will write the error in this approximation as $\mathbf{r} = \mathbf{x} - \mathbf{x}'$

Quality of Fit Measures

We are interested in deciding how well a new patch of image matches the model. Here we will describe several measures of quality of fit, whose performance is compared in the experiments below. If $\mathbf{x}$ represents the grey-levels in the new patch, there are various ways of measuring the quality of fit. In the following the smaller the parameter, the better the fit. Turk and Pentland [2] use a sum of squares of differences between the best model fit to $\mathbf{x}$ and $\mathbf{x}$ itself,

$$R^2 = \mathbf{r}^T\mathbf{r} = (\mathbf{x} - \bar{\mathbf{x}})^T(\mathbf{x} - \bar{\mathbf{x}}) - \mathbf{b}^T\mathbf{b}$$

This only penalises variation in ways not seen in the training set, but doesn't apply limits to the magnitude of variation along modes of the model.

Cootes *et al* [4] use

$$f_0 = M_t + \frac{R^2}{0.5\lambda_t} \qquad\qquad M_t = \sum_{i=1}^{i=t} \frac{b_i^2}{\lambda_i}$$

M_t is a measure of the distance from the mean in the dimensions spanned by the model, and λ_i the i'th largest eigenvalue in the PCA (equal to the variance of the i'th parameter over the training set).

This penalises examples far from the mean along the modes of the model, but the relative weighting between M_t and R^2 is based on the assumption that the sum of the squares of the residuals are approximately gaussian distributed with a variance of $0.5\lambda_t$.

A more sensible approach is to estimate the variance of the sum of the squares of the residuals, V_r, rather than assuming it. Haslam *et al* [7] show how this can be deduced from miss-N-out experiments (or bootstrap/jackknife estimations [6]) using the training data. A quality of fit measure is then

$$f_1 = M_t + \frac{R^2}{V_r}$$

A similar measure is determined from optimal estiimates of the probability distributions by Moghaddan and Pentland [3]. These measures treat all errors from the best model fit, r_j, as equally important, which is not always the case. Errors in poorly modelled pixels are less important than errors in well modelled pixels. By doing miss-N-out experiments we can estimate v_j, the variance of r_j, the residual for the jth element of **x** across the training set. The smaller v_j the better element j is modelled. This information can be incorporated in a new quality of fit measure;

$$f_2 = M_t + \sum_{j=1}^{j=n} \frac{r_j^2}{v_j}$$

Our statistical model of the data is thus

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Pb} + \mathbf{r}$$

where the elements of **b** are zero mean gaussian with variance λ_i, the elements of **r** are zero mean gaussian with variance v_j and the columns of **P** are mutually orthogonal. This is the form of a Factor Model [8], with **b** as the common factors and **r** as the errors.

Model Building

We create models to represent the appearance of the component in each level of a Gaussian Image Pyramid [9]. Thus, given a set of s examples marked on different images, the model building procedure proceeds as follows;

1. For each example extract five grey level patches from image. One at the given position, and four at positions displaced by a fraction of a pixel in x and y. This blurs the model slightly, leading to a more robust model. (Experiments suggest a displacement of ±0.25 is suitable – see below).
2. Apply a Principle Component Analysis to obtain the mean, $\bar{\mathbf{x}}$, and t principle modes of variation **P**. The value of t is chosen so that the model represents a suitable proportion of the variation in the training set (eg 95%) [4].
3. Perform miss–N–out training to estimate the variance of residual errors for each pixel, v_j. (Usually miss–5–out).
4. Repeat at each resolution of a Gaussian Image Pyramid

Search

Suppose we wish to locate the single best location of a model given a region to search in a new image. We can assess the quality of fit of the model at a given point – not necessarily on the pixel grid – by sampling the grey–levels (using bilinear interpolation) in a region around the point, concatenating the values into a vector, **x**, and using the fit measures f_0, f_1 or f_2. Our aim is to locate the point in the search area which minimises the fit measure. To search over a region we use a multi–resolution technique for efficiency, as follows;

1. Project the search region into the coarsest resolution image.
2. Assess the quality of fit at all pixel grid points inside the search area, and choose the best.
3. Use refinement techniques about this point to achieve sub–pixel location. (Resample the image using bilinear interpolation and test the fit).
4. Project current best fit point into a finer resolution image and search over a 3 x 3 grid about this point (experiments with different grid sizes suggest this is optimal).

5. Repeat until sub–pixel location achieved at finest resolution

It is important to locate minima to sub–pixel accuracy, because their widths are often of the order of a single pixel. Without accurate location of the pit, the estimate of the height of the pit is poor, and can lead to poor discrimination of the object from the background (see below).

If the angle of the object in the image is not known, the search can be repeated with the model fixed at different angles, and the one with the best final fit chosen.

Incremental Training

It is possible to use the modelling and search techniques to help a user build a model quickly and efficiently. The procedure is as follows :

1. Draw the outline of the example object in the first image
2. Build a model, using only the given object and four displaced versions.
3. Indicate a region containing the next example
4. Use the current model to locate the best fit in this region.
5. Modify the found location interactively if necessary.
6. Add the new example to the training set and re–build the model.
7. Repeat the process, using the current model to help locate next example.

Usually the system locates most new examples accurately after being shown a small number of examples (sometimes only a single example is required). However, it is important to give as many examples as possible to build the model, to ensure the model parameters are estimated well.

Model Refinement

There will be errors when the examples are marked by hand, or when an early model is used to locate new examples for the training set. One way to reduce these is to use the completed model to relocate each training example. A search area is defined which encloses the existing estimate allowing a suitable border, and the model used to find the best fit to sub–pixel accuracy within this region. Once each training example is so refined, the model is retrained on the refined examples. This can be repeated several times, until the movement of the examples when re–located falls below a certain threshold. Below we show how refining the models in this way can lead to improved performance.

Confirmation of Object Presence

Having located the best fit of a model in a given search region, we wish to confirm that what we have found is the desired target. The quality of fit parameters can be shown to have a particular probability density distribution. If we knew the distribution of fit values for the background or false positive results, we could choose a threshold to minimise a suitable cost weighted Bayes Risk criterion. However the failure modes are generally either unknown or are too varied to model properly. The best that can be done is to model the distribution of correct examples and set a threshold for each model which should fail no more than a given proportion (eg 1 in 1000) of these true examples. Below we demonstrate that this approach will lead to thresholds which should reliably discriminate between good and bad examples even when there is considerable appearance variation for the target structure.

Results of Experiments

The Effect of Model Blurring on Location Accuracy

To investigate the performance of the new algorithm on synthetic data we generated a random test image. We trained a model on a 20 x 20 pixel area about a target point in 20 new images generated by adding gaussian noise to the original image. We trained models both with and without including displaced examples. Both models gave optimal fits either at or close to the original target point. For instance Figure 1 shows a graph of f_0 about the optimum for the model trained without any displaced examples. In this case the trough is narrow – by moving half a pixel away from it the quality of fit falls dramatically. Thus when sampling at pixel intervals, there is a chance we may miss the optimum. Figure 2 shows the fit measure about the optimum for a model trained with examples displaced by 0.25 pixels from the target. The trough has been broadened significantly, and moving half a pixel from the optimum has little effect on the quality of fit.

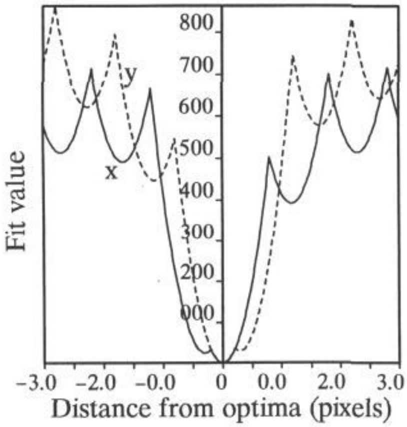

Figure 1 : Cross–section (in x and y) through optimum (model trained without displacement).

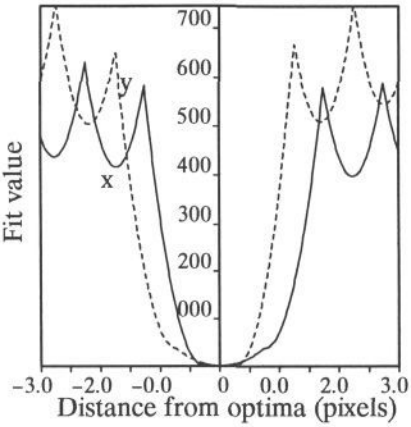

Figure 2 : Cross–section (in x and y) through optimum (model trained with examples displaced by 0.25 pixels).

However, by 'blurring' the model in this way we introduce more uncertainty in the location of the optimum. When a displacement of 0.25 pixels was used, the mean error in optimum position was 0.08 pixels. Since we are unlikely to be able to achieve real accuracy better than 0.1 pixels because of calibration and pixelation considerations, this was considered acceptable.

Effect of Sub–Pixel Refinement

We used the multi–resolution search method described above to locate the best fit in a 40 x 40 area about the true target. To simulate different pixelations the initial search grid was displaced by –0.4,–0.2,0.0, + 0.2, + 0.4 pixels in both x and y (25 tests in all). The distributions of final fit measures (using f_0) and location errors were examined.

We performed the experiment both with and without sub-pixel location of the targets, using blurred and unblurred models. Table 1 summarises the results.

	Unblurred		Blurred (0.25 pixels)	
Sub-Pixel Location	Mean Location Error	Fit Values mean ± sd	Mean Location Error	Fit Values mean ± sd
No	0.36	50 ± 20	0.33	13 ± 8
Yes	0.0	3.1 ± 0.0	0.0	2.9 ± 0.0

Table 1 : Effect of using sub-pixel location and blurring model. (Location error in pixels)

Without sub-pixel refinement there was a large variation in the fit values and a considerable reduction in mean quality of fit caused by failure to find the optimum accurately. The effect occurs for both blurred and unblurred models.

The Distribution of Fit Measure Values

In order to study the performance of the various models and fit measures we tested the approach on images from a printed circuit board inspection problem. We built a model of a region of a capacitor (Figure 3). This model is used in an inspection system to confirm that the capacitor is present and has the correct location and orientation. The model was trained on 12 examples. We used the approach described above to locate 260 capacitor examples in new images of circuit boards. We then studied the distributions of fit values given by different measures. Figure 4 shows the distribution of f_2 fit values for the capacitor model.

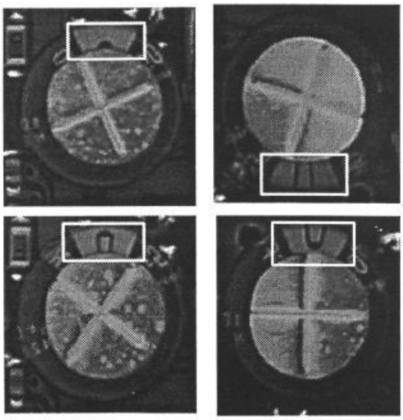

Figure 3 : Typical capacitor examples showing region modelled

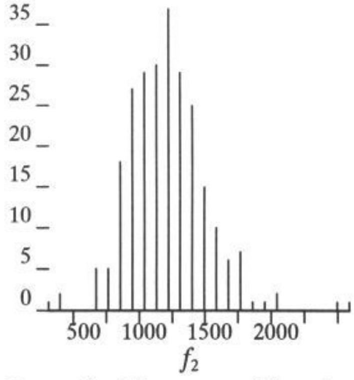

Figure 4 : Histogram of fit values for capacitor model.

If the variation and noise was purely gaussian, we would expect the measures f_1 and f_2 to be distributed as chi-squared of degree *n*, with mean of *n* and a variance of *2n*. In our results, although the mean is about right, the variance appears much larger

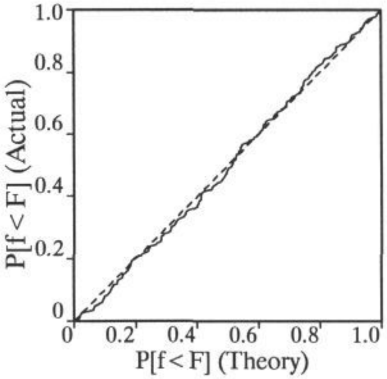

Figure 5 : Comparing estimated with actual p.d.f. for capacitor model.

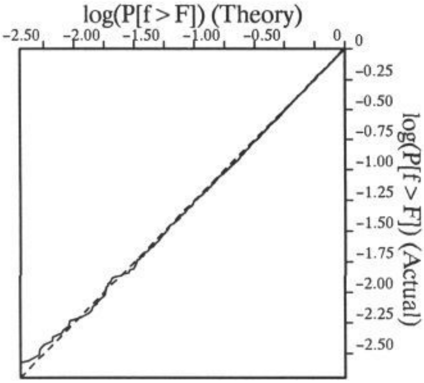

Figure 6 : Comparing model with actual p.d.f. for SMD model, assuming a chi-square shaped distribution.

than $2n$. This suggests that the variation is not truly gaussian, or that the models are not fitting correctly.

In order to use the fit values to estimate the probability of fit, we need a model of this distribution. The distributions across the test set have a chi-square shape, and we have found that a reasonable approximation to this is given by a suitably scaled chi-square distribution with k degrees of freedom, $p(f) = (n/k)X^2(kf/n, k)$. This distribution has a mean of n and a variance of $2n^2/k$. We set $k = 2(n/\sigma)^2$, where σ is the standard deviation of the measured distribution. This has the correct mean and variance. In practice σ could be estimated from a verification set after training. Figure 5 shows the comparison between this estimated distribution and the measured distribution of f_2 for the capacitor model, indicating that the model distribution fits well. Figure 6 shows the comparison between actual and model distributions for fit measure f_2 generated from a set of 3200 examples of a surface mount device (SMD), assuming the chi-square shaped distribution. Assuming a gaussian distribution of fit values leads to a far worse fit.

The Discriminating Power of Different Approaches

We wished to assess how well the models could discriminate between true examples and anything else that might appear. By testing the models on large numbers of images we were able to build a set of 130 false examples which were likely to give good fit values and thus fool an automatic inspection program. These included patches of bare board, damaged components, badly misaligned components and different components with similar appearance to the target. We used the search methods to locate the best fit of the models to these false examples, and compared their fit values with those of our test set of good examples, when located similarly. We tried both with and without a) blurring the training set, b) sub-pixel location and c) refining the training set. We also tried using both f_1 and f_2 as the quality of fit measure. Table 2 summarises the results of the experiments. We show two measures of discrimination.

One is the ratio of the best false example (*BF*) to the worst true example (*WT*). The larger this is, the wider the separation. We also give *Pr[f>BF]* an estimate of the probability that a random true example would be worse than the best false example, assuming the p.d.f. given above.

The table shows that by refining the model we gain considerable improvements in discrimination, and that fit measure f_2 performs considerably better than f_1. Because the R^2 term dominates the M_t term f_0 is approximately a scaled version of f_1 and discrimination results for f_0 are similar to those for f_1. Blurring the model has a smaller effect, but does lead to some improvement. In practice blurring is particularly important when small numbers of training examples are used (for instance during the initial generation of the training set).

When sub-pixel location is used, the quality of fit measures improve more on the bad examples than on the true examples, leading to a reduction in the separation between the distributions. The mean and s.d. of the distribution of fits to good examples both reduce by about 3%, as spurious variation caused by poor location is reduced.

Refined Model?	*Fit Measure*	*Sub-Pixel Location?*	*Blur Model?*	*BF/WT*	*Pr[f>BF]*
Yes	f_2	*Yes*	*Yes*	*2.43*	*1.0e-12*
No	f_2	*Yes*	*Yes*	*1.42*	*4.7e-6*
Yes	f_1	*Yes*	*Yes*	*1.04*	*720e-6*
Yes	f_2	*No*	*Yes*	*2.54*	*0.2e-12*
Yes	f_2	*Yes*	*No*	*2.13*	*30e-12*
Auto-correlation with mean				*0.57*	*0.12*

Table 2 : Effects of varying each parameter in turn on discrimination between fits to true and false capacitor examples (See text for key).

Table 2 shows the results of the same experiment run using standard auto-correlation with the mean patch. There is a considerable overlap between true and false examples, suggesting that the additional information used in the statistical models is essential for successful location and discrimination of components.

Discussion

The multi-resolution search technique described above allows us to locate objects rapidly and accurately. The time to locate each component depends on the number of pixels in the model and the search area, but it takes less than a second on a Sun Sparc20 to find a typical example on a PCB. Although locating targets to sub-pixel accuracy appears to reduce the discrimination between good and bad examples, this accuracy of measurement is often required in an application. It also 'tightens up' the distributions of fit measures by removing spurious variation caused by location errors, so can improve the model of the p.d.f.

It is important to have reliable models of the p.d.f.s for the fit measures, as this allows us to set thresholds which will fail predictable numbers of true examples. We have shown that the p.d.f. is well modelled by a scaled chi-squared distribution whose parameters can be estimated from a test set.

The more comprehensive modelling of the residual errors used in the measure f_2 compared to those used in f_0 or f_1 gives a considerable improvement in the ability of the system to discriminate between true and false examples.

The statistical models give significantly better results than standard auto-correlation with a mean template. Experiments on other components give results comparable with those described above.

Conclusion

The combination of statistical models, new model building methods and sub-pixel multi-resolution search techniques described in this paper lead to a powerful feature detector, and allows thresholds to be chosen based on well modelled probability distributions. By training on a small number of examples we can build tools which allow us to accurately locate and identify objects whose shape varies little but whose grey-level appearance can vary considerably. This approach is particularly useful for industrial inspection, but may have many applications in interpreting medical images and solving other computer vision problems.

References

[1] D.H.Ballard, C.M.Brown, Computer Vision. *Prentice Hall*, 1982.

[2] M. Turk, A. Pentland, Eigenfaces for Recognition. *Journal of Cognitive Neuroscience* Vol.3, No.1 1991. pp.71–86.

[3] B. Moghaddam, A.Pentland. *in Proc. 5th Int. Conf. on Computer Vision,* 1995, pp. 786–793.

[4] T.F.Cootes, A.Hill, C.J.Taylor, J.Haslam, The Use of Active Shape Models for Locating Structures in Medical Images. *Image and Vision Computing Vol.12, No.6 July 1994, 355–366.*

[5] T.F.Cootes , C.J.Taylor, A.Lanitis, Active Shape Models : Evaluation of a Multi-Resolution Method for Improving Image Search, *in Proc. British Machine Vision Conference*, (Ed. E.Hancock) BMVA Press 1994, pp.327–338.

[6] B.Efron and R.Tibshirani, Bootstrap Methods for Standard Errors, Confidence Intervals, and Other Measures of Statistical Accuracy. *Statistical Science*, Vol.1, No.1, 1986, pp54–77.

[7] J.Haslam, C.J.Taylor, T.F.Cootes, A Probabalistic Fitness Measure for Deformable Template Models, *in Proc. British Machine Vision Conference*, (Ed. E.Hancock) BMVA Press 1994, pp.33–42.

[8] R.A.Johnson, D.W. Wichern, Applied Multivariate Statistical Analysis. *Prentice-Hall* 1982.

[9] P.J.Burt, The Pyramid as a Structure for Efficient Computation. in *Multi-Resolution Image Processing and Analysis*. Ed. Rosenfield, pub. Springer-Verlag. 1984. pp. 6 – 37.