

# Contextual Classification of Cracks

N. Bryson, R.N. Dixon, J.J. Hunter, C. J. Taylor

Department of Medical Biophysics,  
Manchester University, Oxford Road, Manchester M13 9PT

## ABSTRACT

We describe a technique for improving the classification of fragmented cues for cracks. Evidence propagation on Bayesian networks represent search within the context of each cue. The algorithm was applied to a data-set of cracks, and results demonstrate that contextual classification of the cues leads to significantly improved error rates.

## 1 Introduction

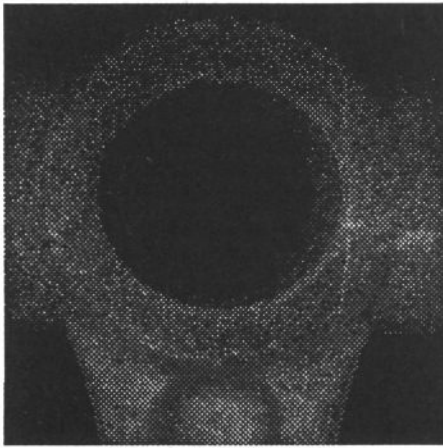
Inspection of forged steel components for cracks is a tedious but necessary process. Several techniques for visualising defects are available, but all produce false indications. It is most cost-effective to inspect after forging and before machining, although the surface has a pitted texture which increases the density of false indications. Manual inspection of large number of components under factory conditions is error-prone, due to fatigue and subjective factors. Several attempts have been made to automate the inspection process, particularly for machined components (for a review, see Bryson et al. 1991). None of the cue-generators so far developed perform perfectly. They generate noise, in the form of false cues, and genuine cues are frequently fragmented, making them hard to discriminate. Some researchers have attempted to overcome this by re-joining or grouping fragmented cues. These results indicate that techniques which allow weak indications to support each other on the basis of proximity, collinearity, etc. will lead to improved classification. This can be described as classification of the cues in the context of neighbouring cues.

We shall describe a general method for doing this, based on Pearl's algorithm for probabilistic reasoning about the values of random variables with constraints. In section 2 we describe a dataset of crack-containing images and a cue-generating process. In section 3 we describe a network of nodes connected with links, where each node is associated with the end of a cue. In section 4 we associate with each link grey-level and shape evidence, which is used to carry out a local, context-free classification of the link. We associate shape evidence with each node, which is used to provide contextual support for the links on either side. Section 5 describes a Bayesian network representing statistical inferencing about the status of a cue and a join between two cues. In section 6 we relate this to local, context-free classification of a single cue. Section 7 shows how the Bayesian network classifies a join between two cues. In section 8-10 we describe how this is used to guide the search for the optimal explanation of the status of cues and joins in the context of the status of cues and joins in its neighbourhood. The results of applying the method to a data-set of one hundred images of cracks on the small end of connecting rods are then presented in section 11, and discussed in section 12.

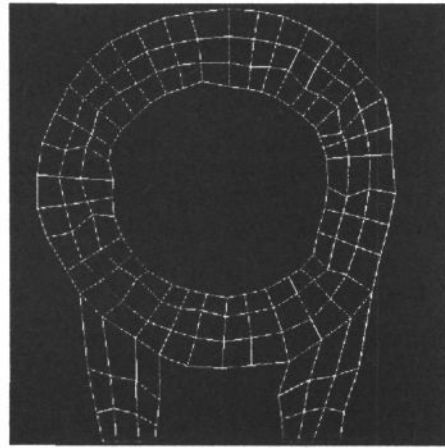
## 2 Cue Generation

The exemplar used in this paper consists of a set of connecting-rods for automotive engines. A set of components which had been rejected by human inspectors was col-

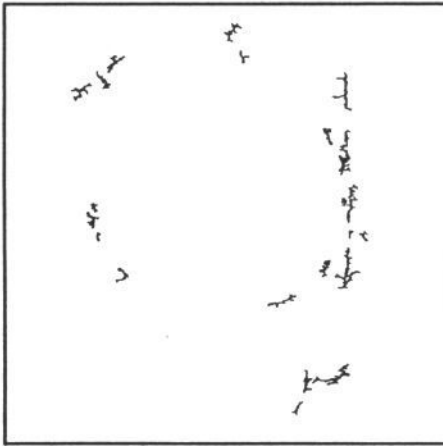
lected, de-magnetised, cleaned, remagnetised, bathed in magnetic ink, and imaged under UV light. The images were collected using a standard CCD camera, using a rig which held the component in a fixed position relative to the camera. A typical source image is shown in figure 1(a). The surface to be inspected was sub-divided using a mesh shown in figure 1(b).



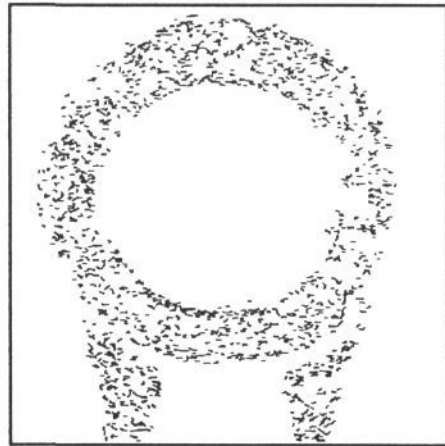
(a) Source image



(b) Mesh



(c) Blobs labelled as cracks



(d) Blobs labelled as non-cracks

*Figure 1 – Output of the cue-generator*

The first step in cue-generation is to remove, by adaptive thresholding, all pixels which are too dark to be of interest. A grey-level histogram is taken over each local mesh area, and a threshold selected to reject 50% of pixels in the area.

The next step is to apply adaptively a line operator to each pixel which survived the previous step. This measures the contrast between pixels lying along each of eight lines through a 5x5 neighbourhood centred on the pixel, and the rest of the neighbourhood, and selects the direction with maximum contrast. A histogram of the re-

sponses within the local mesh area is generated, and a threshold on the contrast selected which passes 10% of the pixels to the next stage.

We now extract blobs from the binary image of surviving pixels, and generate a feature vector for each blob. The feature vector consists of the length (maximum chord length), mean grey-level, and orientation. Very short blobs ( $< 3$  pixels) are then rejected. The orientation of the candidate was used because, in this dataset, the cracks were primarily due to thermal cracking along the length of the original steel billet.

The remaining blobs are classified on the basis of length, and orientation, using a combination of linear classifiers. Short blobs were required to be nearly vertical, whereas long blobs could have any orientation. Figure 1(c) shows the surviving blobs from the source image in figure 1(a), with the blobs which had been rejected shown in figure 1(d). The result is typical, in that the crack has been fragmented, and some noise cues have survived.

### 3 The Join Network

The cues are then converted to splines, and candidate joins interpolated between the end-points, with continuity of gradient enforced. This generates plausible 'coarse' representations of the candidates. A sparse join network is then constructed by restricting the number of candidate joins at each node. A maximum of three joins per node was adequate for this dataset. The joins are selected on the basis of proximity, and joins longer than either of the cues are disallowed. This ensures that the number of joins was of the same order as the number of cues.

### 4 The Evidence

The evidence used in the local reasoning process about cues consists of (a) the mean grey-level of the cue (b) the length of the cue (c) the orientation of the cue.

A join between two cues is more likely to be valid if it is short, and the resultant shape is approximately linear. We evaluate a linearity measure at each end of the join (i.e. at each node), by carrying out a principal components analysis of the distribution of pixel positions of the candidate shape within a disc centred on the node. The ratio of the eigenvalues of the covariance matrix of the point distribution gives a measure of linearity.

### 5 Bayesian Reasoning about a Join between Cues

The reasoning process about the status of a link between the ends of two cues, given the evidence described above, can be represented by the Bayesian network [2, 3], shown in figure 2. The boxes represent random variables, and the arrows represents a causal dependency of the state of belief about the variable at the head of the arrow on the state of belief about the variable at the tail of the arrow. The variables  $L_{i1}^{i2}$ ,  $L_{k1}^{k2}$  and  $L_{i2}^{k1}$  represent the possible labellings of the cues and the link between them. The sub-nets surrounded by dotted lines provide prior and evidential contextual support for the label  $L_{i2}^{k1}$  on the join, described below. The evidence variables  $g_{i1}^{i2}$ ,  $l_{i1}^{i2}$  and  $a_{i1}^{i2}$  represent the mean grey-level, length and orientation, respectively, associated with a link, while  $N_{i2}$  represents the local eigenratio evaluated at a node at one end of a join. The arrow connecting  $L_{i1}^{i2}$  to  $L_{i2}^{k1}$  represents the con-

straint that a join can only be made to a crack. The arrows connecting  $N_{i_2}$  to  $L_{i_1}^{i_2}$  and  $L_{i_2}^{k_1}$  represent the constraint on the distribution of the eigenratio values imposed by the labels on the links.

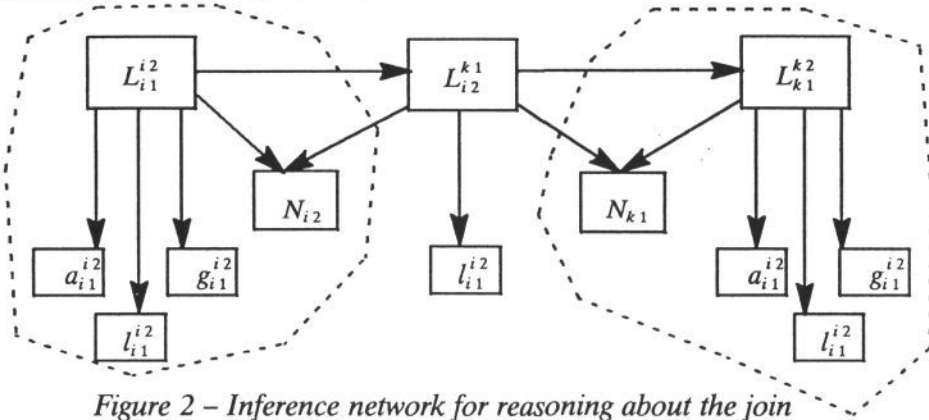


Figure 2 – Inference network for reasoning about the join between two cues shown.

We assume that we can acquire from training, or estimation, the probability distributions:  $p(L_{ij}^{kl})$  – the prior probability distribution for a label;  $p(g_{ij}^{kl} | L_{ij}^{kl})$  – the distribution of mean cue brightness for cracks and non-cracks;  $p(l_{k_1}^{k_2} | L_{k_1}^{k_2})$  – the distribution of lengths of crack and non-crack cues;  $p(a_{k_1}^{k_2} | L_{k_1}^{k_2})$  – the distribution of orientations of crack and non-crack cues;  $p(l_{i_2}^{k_1} | L_{i_2}^{k_1} = crack)$  – the distribution of lengths of valid joins between cracks;  $p(l_{i_2}^{k_1} | L_{i_2}^{k_1} = non - crack)$  – the distribution of lengths of non-valid joins;  $p(N_{i_2} | L_{i_1}^{i_2} = crack, L_{i_2}^{k_1} = crack)$  – the distribution of local eigenratios at nodes where a valid join is made;  $p(N_{i_2} | L_{i_1}^{i_2} = non - crack or L_{i_2}^{k_1} = non - crack)$  – the distribution of local eigenratios at nodes where a non-valid join is made. Suitable distributions, modelled on histograms from training data, were generated using Gaussian functions.

## 6 Local Classification of cues

Each cue is classified purely on the basis of the local evidence, by evaluating  $p(L_{i_1}^{i_2} | l_{i_1}^{i_2}, g_{i_1}^{i_2}, a_{i_1}^{i_2})$  using the above distributions with Bayes theorem and selecting the label  $L_{ij}^{kl}$  from the set {"crack", "non-crack"} which gives the maximum.

## 7 Classifying an individual join

We now attempt to classify a join in the context of the evidence associated with the two cues at either end. The optimal value of  $L_{i_2}^{k_1}$  is chosen by generating contextual prior and evidential support functions,  $\Pi_{i_2}^{k_1}(L_{i_2}^{k_1})$ ,  $\Lambda_{i_2}^{k_1}(L_{i_2}^{k_1})$  from the rest of the

network, and then combining these with the local prior and evidential support functions of the join,  $\pi_{i_2}^{k_1}(L_{i_2}^{k_1})$  and  $\lambda_{i_2}^{k_1}(L_{i_2}^{k_1})$ , to provide the total support for each of the labels  $L_{i_2}^{k_1}$ . The  $\Pi$  and  $\Lambda$  functions are generated using Pearl's algorithm. For reasons which will be explained below, it is advantageous for contextual classification to also calculate at this point constrained support functions representing the amount of support provided for the join when the cue propagating the support is constrained to be labelled "crack".

## 8 Contextual Classification

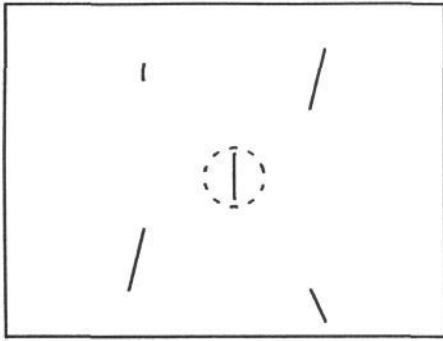
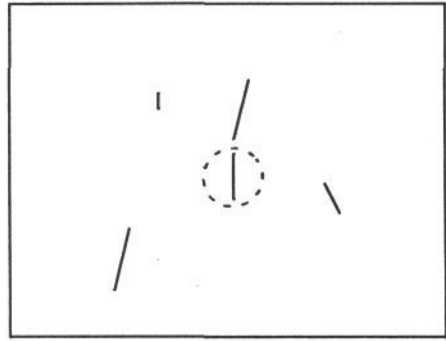
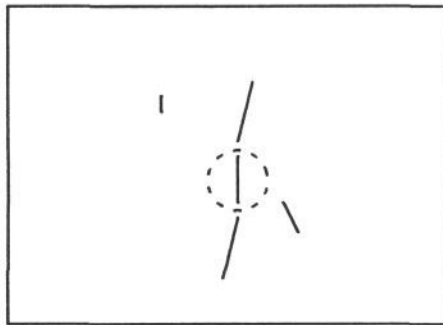
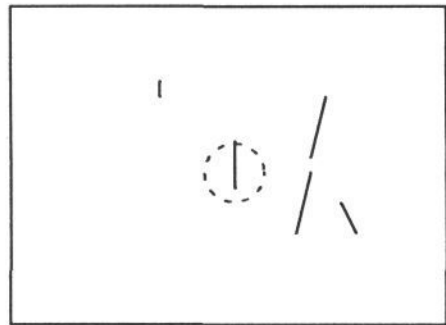
If our cue-generation process generated only two cues, the above method would be an optimal method for selecting labels for the two cues and the join. Because the join network in practice, becomes multiply-connected, the search for the most supportive configuration of the context for each cue cannot easily be broken down into sub-searches, and it is difficult to re-use the results of each such search within other contexts. Singly-connected networks allow the decomposition of the search as described above. Pearl's algorithm is efficient because it allows the context for each decision to be decomposed in a way which allows maximum re-use of searches carried out at each decision node. When the net is multiply-connected, sub-nets are separated from the rest of the net by more than one decision node, and so the search-space defined by the sub-net must be searched several times, with different constraints, to provide support for each decision node.

We approximate the search through the configurations of each context by reducing the connectivity of the context, and by searching likely configurations which form singly-connected networks. We rely on the fact that although the neighbourhood network is multiply-connected, the network of crack fragments tends to be singly-connected. When, for a given fragment, we select a physically likely context, we can then assume that most of the neighbourhood net links in its context can be ignored (since the physical context forces them to be invalid). The sub-nets are then singly-connected, and so we can re-use the results of local searches.

There are four physically realizable contexts for a cue likely to arise in practice, as shown in figure 3: the cue is independent of its neighbours (i); the cue is at the end of a crack (e); the cue is part of a crack (p); the cue is an isolated non-crack near a crack (n).

We assume as few joins as possible i.e. none in context (i), one in contexts (e) and (n), and two in context (p). Evaluation requires the combination of  $\Pi$  and  $\Lambda$  functions, which have already been evaluated in the previous stages, described above, so that relatively little computation is required. The solution method, in outline, is as follows.

1. Local prior and evidential support functions are evaluated for each cue, providing a local classification, as described in section 6 .
2.  $\Pi$ ,  $\Pi^*$ ,  $\Lambda$  and  $\Lambda^*$  functions for the joins are calculated as described in section 7 . This provides a contextual classification of each join.
3. For each of the possible label states of each cue in its approximate context, we seek the physically realizable context which provides maximum support. This is sketched in section 9 .

(i) *No support between cues*(e) *cue is at end of a crack*(p) *cue is part of a crack*(n) *cue near a crack**Figure 3 – Physically realizable configurations of context*

4. Choose the label with maximum support.
5. Update the  $\Pi$  and  $\Lambda$  functions provided by each cue.
6. Repeat steps 2–5 until the state of belief at each cue ceases to change.

At the highest level the method is similar to the "iterated conditional mode" (ICM) method of Besag [4], in that at each iteration, each decision site selects the most probable label, given the current set of labels on its neighbours. The difference is that each site keeps track (via the  $\Pi$  and  $\Lambda$  functions) of from where its support comes.

## 9 Comparison of Contextual Configurations

The total state of belief at each cue, within its context, consists of a product of terms of the form  $p(\text{evidence} \mid \text{label}(s))$ . When comparing configurations, it is unnecessary to evaluate all such terms. The only terms which must be evaluated are those which change between configurations. We need only compare changes from the (i) context configuration.

## 10 Updating the Contextual Support Functions

Now that each cue has received contextual evidence, and has decided on the most probable configuration of its neighbourhood, it can now update the support functions it provides to its neighbours. The outgoing  $\Pi$  and  $\Lambda$  functions are modified using the incoming constrained support functions, as appropriate. This ensures that the outgoing support functions reflects the evidence upon which the current labelling of the cue is based.

## 11 Results

When applied to the cues generated in figure 1(c), the algorithm led to the results shown in figure 4, with  $p(L_{ij}^{kl} = crack)$  set equal to 0.08. The images show the cues which have been labelled "crack" at the end of the first two iterations (the joins have not been shown, for clarity). After two further iterations, the balance of beliefs on each cue changed, but without changing the label with maximum support. The cue labelled FP in figure 4(b) is a false positive caused by a scratch on the component, which could easily be mistaken for a crack. A genuine cue at the position marked FN did not get labelled correctly, because its shape did not align well with the nearby true positive. All the other cues were true positives, with increased odds-ratios as a result of the contextual support.

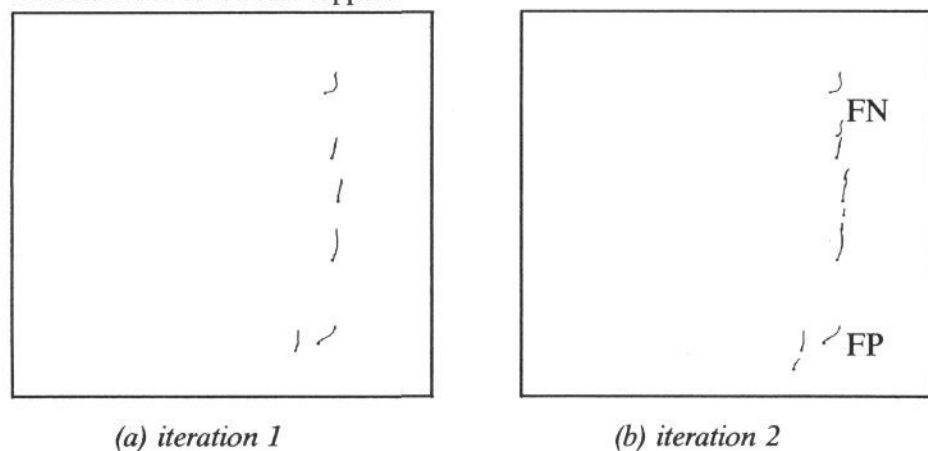


Figure 4 – Output of first two iterations.

The algorithm was applied to one hundred images, chosen at random. The dataset yielded 265 genuine crack cues, and 1440 non-crack cues. The Receiver Operating Characteristic (ROC) curve, showing the locus of the true positive rate (TP) and the false positive rate (FP) was generated by varying the prior probability of a cue being a crack ( $p(L_{ij}^{kl} = crack)$ ) between 0.02 and 0.2. This effectively provides a threshold for the weight of support required to label a cue as a crack. One ROC curve shows the results for the local classification, while the other shows the results for the contextual classifier. The sample success and failure rates were used as estimators for the population success and failure rates, with 95% confidence interval error bars [5] as shown. The results indicate a significant improvement in classification from using context.

For comparison, we have also derived a linear discriminant rule. A set of 178 features were measured for each of the combined candidates, and the best nine selected

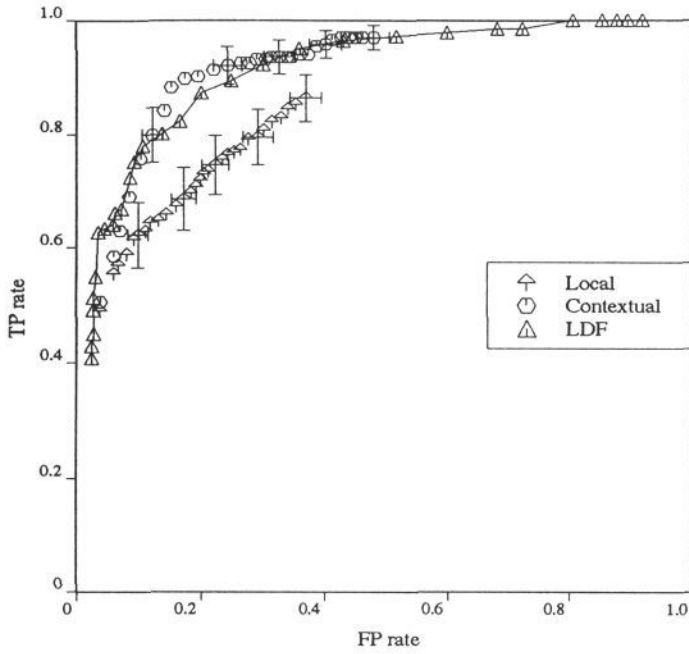


Figure 5 – ROC curves for local and contextual classification, contrasted with the ROC curve of a standard linear classifier

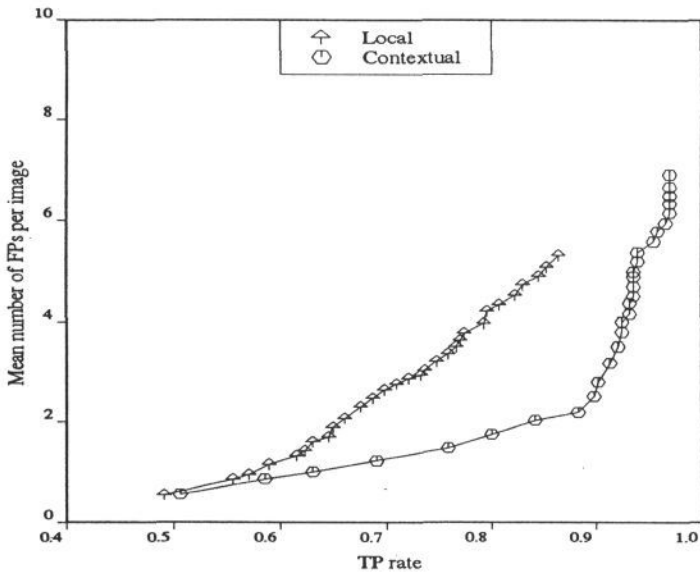


Figure 6 – Mean number of false positives per image v. TP rate for local and contextual classifiers.

for use in an optimal discriminant rule. The results encourage us to believe that, for this application, the use of a small set of local features, combined with contextual constraints, outperforms classifiers based on global features.

The use of a contextual classifier increases the false positive rate, because some noise cues may be "promoted" by being near other noise, or a genuine cue. However, the price is worth paying, as shown in figure 6. For a given TP rate, the contextual classifier gives a smaller mean number of false positives per image.

## 12 Discussion

Context has long been used in machine vision to support classification [6,7,8]. These methods are based on the use of functions measuring the compatibility of a pixel label with the labels attached to its neighbours. Generally, the compatibility function measures pair-wise compatibility between the pixel label and a neighbouring pixel. Kittler and Hancock [9] have recently extended this approach by introducing a dictionary of permissible context labellings, which allows more complex constraints to be expressed, without leading to a computational explosion. The set of physically realizable configurations of context (illustrated in figure 3) corresponds to this dictionary. However, Kittler and Hancock were dealing with pixel-based labelling, with a fixed neighbourhood structure for all pixels, and so could maintain an explicit dictionary of permissible neighbour configurations. In our case we deal with blobs, and must compute the neighbourhoods and support functions at run-time.

A common failing of probabilistic relaxation methods is that support coming from a neighbour is fed back to the neighbour on the next iteration. We have exploited the ordering imposed by the linear structures we seek to keep track of the source of support, via the  $\Pi$  and  $\Lambda$  functions. Our method for updating the  $\Pi$  and  $\Lambda$  functions ensures that support is fed consistently back into the context, preventing cycles of positive feedback.

There are several avenues available for developing the technique. New joins could be postulated after each iteration, using the joined cues to provide a new length scale. After relaxation, further evidence for the joins could be sought by adding to the join-network non-surviving cues in the vicinity of joins, or ends of cues. Problem-specific knowledge could be brought to bear by using different sets of conditional probability distributions in different parts of the network. An expanded set of labels can also be used, to allow more specific classification of "non-cracks".

## 13 Conclusions

We have shown how Pearl's algorithm for belief updating on a singly-connected Bayesian net can be extended to approximate the search for the globally-optimal labelling of a multiply-connected net when we are constrained to seek linear structures. The method has been used to improve the classification of cues for cracks embedded in noise on the rough surface of forged components.

## 14 References

- 1 "A survey on optimisation and automation of FMPI", Bryson N.F., Dixon R.N., Hunter J.J., Taylor C.J., Internal Technical Report, Wolfson Image Analysis Unit, Dept. of Medical Biophysics, University of Manchester, 1991.
- 2 "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference", Judea Pearl, Morgan Kaufmann Publishers, Inc., 1988

- 3 "Boundary Detection using Bayesian Nets", N. Bryson, C.J. Taylor, *Image and Vision Computing*, Vol. 10, No. 5, June 1992, pp. 308-312
- 4 "On the Statistical Analysis of Dirty Pictures", Besag J., *J. Roy. Statist. Soc. B* (1986), Vol. 48, No. 3, pp. 259-302
- 5 "Introduction to Statistical Inference", E.S. Keeping, pp.114-115, Van Nostrand (1962)
- 6 "Scene Labelling by Relaxation Operations", A. Rosenfeld, R.A. Hummel, S. W. Zucker, *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 6, No. 6, pp. 420-433, June 1976
- 7 "An Application of Relaxation Labelling to Line and Curve Enhancement", S.W. Zucker, R.A. Hummel, A. Rosenfeld, *IEEE Trans. on Computers*, Vol. C-26, No. 4, pp. 394-403, April 1977.
- 8 "Relaxation labelling algorithms - a review", J. Kittler, J. Illingworth, *Image and Vision Computing*, Vol. 3, No. 4, pp. 206-216, November 1985
- 9 "Edge Labelling using Dictionary-based Relaxation", Hancock E.R., Kittler J., *IEEE PAMI*, Vol. 5, No. 2, pp. 165-181, Feb. 1990