Optimal probabilistic relaxation labeling

Ian Poole

Medical Research Council
Human Genetics Unit
Western General Hospital
Edinburgh EH4 2XU *

This paper investigates the theoretical limits of probabilistic relaxation labeling (PRL), applied to per-pixel contextual image classification. The performance of a scheme which is defined *to be optimal (within a class of PRL schemes) is studied, and found to fall short of that theoretically obtainable by directly considering all the original* a posteriori *probabilities (PPs) in the image. It is also found that an optimal scheme must use different updating functions at each iteration, and that these functions will depend on the distributions of the original per-pixel data.*

An estimation based implementation of the optimal scheme — termed 'trained probabilistic relaxation' (TPR) is then described, which, in spite of it's theoretical limitations, has a number of commendable characteristics. Experimental results are presented.

We are concerned with the problem of obtaining a per-pixel classification of an image, such as is common in remote sensing where a land-use thematic map is to be automatically derived from multispectral data. Another example is the labeling of 'edge' pixels as the first stage of a machine vision system. It is widely accepted that classification results can be considerably improved when contextual information around a pixel is taken into account, that is, when the classification of each pixel is influenced by pixels at a (possibly considerable) distance from the one in question.

To fix notation, assume that with each pixel i is associated the random variables Y_i and X_i, representing the pixel's true *class* (or label) and *data* respectively. Realizations of Y_i and X_i for a particular image will be written y_i and x_i respectively. For example, in a remote sensing context, each y_i might take on values from the set $\Phi = \{forest, water, other\}$, with x_i a vector representing the measured intensities in a number of spectral bands.

Given a suitable image with a pre-classified training overlay, many methods exist for estimating the class-conditional distributions of the image pixel data — ie $\mathbf{d}(X|Y{=}\alpha)$ for each α in Φ, and the class prior probability vector $\mathbf{D}(Y)$. Here $\mathbf{d}$ indicates a density function over the uninstatiated random variable (X), and $\mathbf{D}$ a probability vector. Note that the subscripts may be dropped since we assume the image to exhibit stationary (ie shift independent) statistics. Once these distributions have been estimated it is a straightforward matter to assign an initial *a posteriori* probability vector $\mathbf{q}_i^0, \mathbf{q}_i^0 = \mathbf{D}(Y_i|Y_i{=}y_i)$ to each pixel in subsequent images.

Probabilistic relaxation labeling (PRL) is a means of iteratively updating these initial probability vectors in the light of contextual evidence from neighbours. PRL aims to generate PPs conditioned on a larger window of data and so minimize the proportion of pixels misclassified by a maximum *a posteriori* probability (MAP) classifier by combining the PPs from the central pixel and (say) its 4-connected neighbours. This new PP can be called $\mathbf{q}_C^1$, ie:

$$\mathbf{q}_C^1 = f(\mathbf{q}_C^0, \mathbf{q}_N^0, \mathbf{q}_S^0, \mathbf{q}_E^0, \mathbf{q}_W^0)$$

where the subscript C,N,S,E,W stand for Centre, North, South etc. A variety of updating functions have been proposed (see [5] for a review), and it is typical to reapply the same function in an iterative fashion so that at the kth iteration we have:

$$\mathbf{q}_C^{k+1} = f(\mathbf{q}_C^k, \mathbf{q}_N^k, \mathbf{q}_S^k, \mathbf{q}_E^k, \mathbf{q}_W^k).$$

In the following two sections we investigate the performance limits of this scheme; proofs of theorems are outlined only — see [9] for full details.

The incrementally optimal updating (IOU) scheme — $\mathrm{g}^1 \ldots \mathrm{g}^k$

Much of the theoretical analysis to follow is formally limited to one-dimensional "images"; arguments are

*Work was carried out whilst the author was at the Dept. of Computer Science, University College London, with the support of the SERC and the National Physical Lab., Teddington.

BMVC 1990 doi:10.5244/C.4.18

given in [9] which leave little doubt that the results will also be valid in higher dimensions.

In the 1-D situation, a pixel has only two neighbours, one to the left and one to the right. This is shown in a fig. 1. At the kth iteration, a function, $\mathbf{g}^k$, of

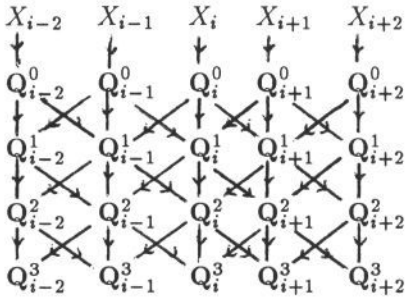


Figure 1: Incrementally optimal updating

three-PPs, generates the new PP for each pixel. The PP vectors, at pixel i, resulting from the first, second, third etc iteration of IOU are written $\mathbf{q}_i^1, \mathbf{q}_i^2, \mathbf{q}_i^3$, etc, with associated random variables $\mathbf{Q}_i^1, \mathbf{Q}_i^2, \mathbf{Q}_i^3$. Traditionally, the update function is identical for each iteration ($\mathbf{g}^k = \mathbf{g}^1\ \forall k$), but we wish to relax this constraint and investigate the performance of a tailored *sequence* of updating functions — $\mathbf{g}^1, \mathbf{g}^2 ... \mathbf{g}^k$. So that:

$$\mathbf{q}_i^k = \mathbf{g}^k(\mathbf{q}_{i-1}^{k-1}, \mathbf{q}_i^{k-1}, \mathbf{q}_{i+1}^{k-1}).$$

We require $\mathbf{g}^k$ to generate an *honest* (see [2]) probability vector, and further that it should be fully refined on the available information. It thus follows that we should define:

$$\mathbf{g}^k(\mathbf{q}_{i-1}^k, \mathbf{q}_i^k, \mathbf{q}_{i+1}^k) =$$

$$\mathbf{D}(Y_i|\mathbf{Q}_{i-1}^k{=}\mathbf{q}_{i-1}^k, \mathbf{Q}_i^k{=}\mathbf{q}_i^k, \mathbf{Q}_{i+1}^k{=}\mathbf{q}_{i+1}^k) \qquad (1)$$

as is shown in the figure. A similar formulation is adopted by Peleg [7].

To say that equation 1 defines an *optimal* updating function is something of a truism; note for example, that it is true *independently* of the assumed image model, ie it holds regardless of whether the data is class-conditionally independent, or the labels exhibit the Markov property. Even if there is texture present, it is still optimal amongst PRL schemes, though the optimum will be a poor one. Note that one may always take equation 1 literally (and analogously for 2D images) and, given sufficient training data, directly estimate the given conditional probability. This will be taken up in shortly.

Theoretical analysis of IOU

The incrementally optimal updating scheme has been *defined* to be optimal — ie a sequence of updating functions tailored to a given image model and iteration, delivering the *a posteriori* class estimate given the local neighbourhood of current PPs. Since the IOU scheme is definitively optimal, anything it cannot achieve cannot be achieved by *any* PRL scheme.

The analysis to be presented is based on the "tooth-comb model" which is described below. Although no closed-form expression has been obtained for the updating functions, even under this restricted model, a number of useful theorems have been developed, and these are presented in the following sub-sections.

The tooth-comb model

This is a simple 1-D model in which the data at each pixel is statistically independent from other pixels given their class, and the contextual dependence between pixel classes may be modeled by a first-order Markov chain. Y_i are the class (random) variables, and X_i are the associated pixel data. The qualitative aspects of the model may be specified as follows: [1]

- $X_i \coprod X_j | Y_i \quad \forall i, j : i \neq j$ (class-conditional independence (CCI));
- $Y_i \coprod Y_j | (Y_{i-1}, Y_{i+1}) \quad \forall i, j : i \neq j$ (the Markov property).

and in fact these two statements completely specify the qualitative aspects of the model.

An instance of the model requires the following to be quantified:

- the set of classes — $\Phi = \{1..m\}$;
- the class priors — $\mathbf{D}(Y_i)$;
- the transition probabilities — $\mathbf{D}(Y_{i+1}|Y_i)$;
- the class-conditional density distributions $\mathbf{d}(X_i|Y_i{=}\alpha)$ for each α in Φ.

Realizations of Y_i and X_i are written as y_i and x_i respectively. Often, the event $X_i{=}x_i$ is abbreviated to simply x_i.

[1] $A \coprod B|C$ reads as 'A is independent of B *given* C', or 'Once we know C, B tells us nothing more about A (or A about B)'

Single-stage, w-reaching context functions — $\mathbf{F}^w$

The benchmark against which any context exploiting scheme may be judged is the function which returns the *a posteriori* probability w.r.t class, given the original per-pixel PPs in as large a window as required. For the tooth-comb model this leads to a family of functions defined as follows:

$$\mathbf{F}^w(\mathbf{q}^0_{i-w} \cdots \mathbf{q}^0_{i+w}) \stackrel{\text{def}}{=}$$

$$\mathbf{D}(Y_i|\mathbf{q}^0_{i-w} \cdots \mathbf{q}^0_{i+w}) \quad (2)$$

Thus for example, $\mathbf{F}^1$ is a function of three PPs and $\mathbf{F}^2$ a function of 5 PPs. For the tooth-comb model, $\mathbf{F}^w$ has a tractable algebraic form which we omit here. For the 2-D case, $\mathbf{F}^w$ may be defined analogously, but to the author's knowledge, no closed form expression is possible without simplifying the model.

Note that $\mathbf{F}^1$ is identical to $\mathbf{g}^1$ by definition.

Do per-pixel PPs preserve all relevant information?

The first stage of any PRL scheme is to generate the *a posteriori* probability vector for each pixel i given its data, ie $\mathbf{q}^0_i \stackrel{\text{def}}{=} \mathbf{D}(Y_i|\boldsymbol{x}_i)$. Once this has been done, the data $\boldsymbol{x}_i$, will not be used again. We would like to know, therefore, whether this transformation from data to PPs discards any information which might be relevant to a subsequent pixel-centred classification. Intuitively we are asking "Could I make as good an estimate of class for each pixel if allowed to see only the PPs for the whole image, as I could if I were allowed to see the *original data* for the whole image?". In fact we can assert the following:

Theorem 1 *When the pixel data is class-conditionally independent, all information relevant to classification is retained by the per-pixel PPs, ie, for any w,*

$$X_i \coprod_{\forall i \neq j} X_j | Y_i \Rightarrow$$

$$\mathbf{D}(Y_i|\boldsymbol{x}_{i-w} \cdots \boldsymbol{x}_{i+w}) = \quad \mathbf{D}(Y_i|\mathbf{q}^0_{i-w} \cdots \mathbf{q}^0_{i+w}) \quad (3)$$

The proof simply involves showing that $\mathbf{D}(Y_i|\boldsymbol{x}_{i-w} \cdots \boldsymbol{x}_{i+w})$ can be expressed as a function of $\mathbf{q}^0_{i-w} \cdots \mathbf{q}^0_{i+w}$ which does not depend on $\boldsymbol{x}_{i-w} \cdots \boldsymbol{x}_{i+w}$; clearly the function must be $\mathbf{F}^w$. This result assures us that having replaced all the $\boldsymbol{x}$s with $\mathbf{q}^0$s we still have a chance of finding $\mathbf{D}(Y_i|\boldsymbol{x}_{i-w} \cdots \boldsymbol{x}_{i+w})$ for any w.

Does the IOU scheme match a one-stage function?

We would like to know whether the sequential application of the IOU functions can achieve equivalent results to the single stage context function defined above. In fact the answer is in the negative, at least for two iterations:

Theorem 2 $\mathbf{F}^2 \neq \mathbf{g}^2.\mathbf{g}^1$.

[2] As has been mentioned, $\mathbf{g}^1$ is simply $\mathbf{F}^1$, so the proof requires it to be shown that $\mathbf{g}^1$ has discarded information relevant to the computation of $\mathbf{F}^2$. This will happen if two or more combinations of $\mathbf{q}^0_{i-2} \cdots \mathbf{q}^0_{i+2}$ values map onto the *same* combination of $\mathbf{q}_{i-1} \cdots \mathbf{q}_{i+1}$ values but to a *different* value under $\mathbf{F}^2$; then clearly *no* $\mathbf{g}^2$ function could be found to satisfy the identity. The mechanics of this proof require a considerable amount of algebraic manipulation and this was carried out with the aid of the computer algebra system REDUCE [10].

The implication of this result is that *no* PRL updating function exists that after two iterations, achieves a probabilistic assessment which is as good as could be achieved by an assessment based on the original data (or, by theorem 1 on the initial per-pixel PPs).

Do the IOU functions depend on the distributions of the original data?

The updating functions $\mathbf{g}^1$, $\mathbf{g}^2$ etc will clearly depend strongly on factors associated with the spatial layout of the classes (the transition probabilities in the tooth-comb model), but intuitively we do not expect the functions to depend on the class conditional distributions of the original data — ie on $\mathbf{d}(X_i|Y_i{=}\alpha)$. Our intuition is supported by the fact that $\mathbf{g}^1$ ($= \mathbf{F}^1$) did not so depend; but what of $\mathbf{g}^2$ and beyond?

Theorem 3 *The IOU function for the second iteration — $\mathbf{g}^2$ depends on the class-conditional distributions of the data for the tooth-comb model.*

The reader is refered to [9] for the proof. This result makes it clear that any attempt to find a closed-form expression for $\mathbf{g}^2$ will be greatly complicated by the involvement of the data distributions. Put another way, a second stage updating function which is optimal for a particular model instantiation, will not be optimal if the data distributions are changed *even though* the spatial (contextual) layout of the classes remain unchanged.

[2] The notation $\mathbf{g}^2.\mathbf{g}^1$ indicates the application $\mathbf{g}^1$ followed by $\mathbf{g}^2$, ie $\mathbf{g}^2.\mathbf{g}^1(a,b,c,d,e) \stackrel{\text{def}}{=} \mathbf{g}^2(\mathbf{g}^1(a,b,c), \mathbf{g}^1(b,c,d), \mathbf{g}^1(c,d,e))$

What of iterating $\mathbf{F}^1$?

A corollary of theorem 3 is that $\mathbf{g}^2$ cannot in general be equivalent to $\mathbf{F}^1$. This follows from the the fact that $\mathbf{F}^1$ is known *not* to depend on the data distributions. It is demonstrated empirically in [9] that the effect of iteratively applying $\mathbf{F}^1$ is to produce PPs which are *optimistic*; that is, the components of the computed PPs rapidly converge to zero or one indicating a high degree of confidance in the MAP classification which is not born out by the achieved classification accuracy. Further, it is also shown that the achieved accuracy is inferior to that obtained by using the correct sequence of IOU functions.

Trained probabilistic relaxation (TPR)

Trained probabilistic relaxation is the name given by the author to a practical implementation of the IOU scheme. TPR involves directly *estimating* the distribution $\mathbf{D}(Y_C|\mathbf{q}_C^k, \mathbf{q}_N^k, \mathbf{q}_S^k, \mathbf{q}_E^k, \mathbf{q}_W^k)$ from training data. A pre-requisite of TPR is the existence of a trainable per-pixel classifier, which generates honest PPs conditioned on the five pixels of a four-connected neighbourhood. There are many ways that such a classifier could be constructed (for example the kth nearest neighbour or parzen estimator — see eg [3]) — the system developed by the author involves the growing of a probability tree, and is summarized in the following sub-section; for a full description see [8] or [9].

The "Lapwing" classifier system

The system extracts a pattern vector for each pixel derived from the pixel data within a local niegbourhood. The size of the window may take on various sizes and shapes — from just the central pixel (whence it may be used to derive initial per-pixel PPs) up to a 9×9 square. During training, the system produces a *probability tree*; each non-terminal node of the tree representing a split in the pattern space. The splits are limited to hyperplanes, but can be in any orientation and are selected on the basis of two main criteria:

- *purity* w.r.t pattern class in the two sub-spaces;
- *robustness* of the split, that is, the extent to which the partitioning hyperplane avoids passing through regions of high sample density.

These criteria are combined to form a single cost function whose optimization is managed by a *genetic algorithm* (see eg [4]). Splitting is continued until a node is either pure, or contains fewer than a preset number of representatives from any class.

The resulting probability tree thus represents a hierarchical partition of the pattern space; inside each partition the class conditional pattern density is assumed to be constant and thus the *a posteriori* probability vector for any point in that region may be determined directly from the normalized ratio by class of the sample members which fall into the region.

Once the tree has been grown it is *pruned* with reference to an independent validation sample. The technique is known as *minimum cost-complexity pruning* and is described in [1]. In essence, the pruning process has the effect of removing splits where either of the two sub-regions have captured too few sample members, or where the class-conditional densities on either side of the split are similar and so the split achieves little.

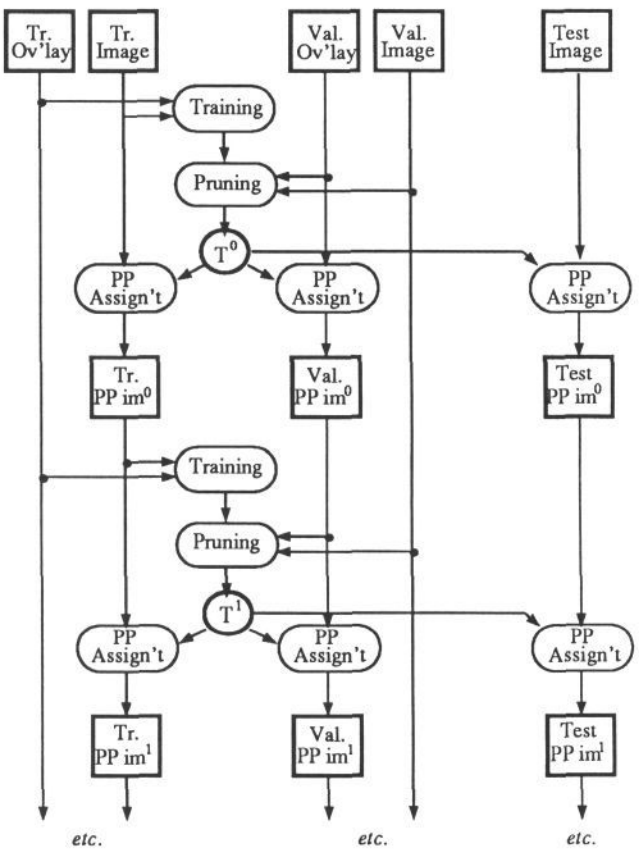


Figure 2: Trained probabilistic relaxation (TPR)

Figure 2 shows the overall scheme of TPR. Two example images with pre-classified training overlays are required for training and validation; these are indicated as 'Tr.' and 'Val.' in the figure.

The first tree, T^0, is trained on the original image data using a neighbourhood involving the central pixel only

[3]. This tree, after pruning may then be used to generate a "probability image" from the training and validation images. For two class problems, a probability image may be represented by single grey-level image, where the intensity at the ith pixel is proportionally related to $P(Y_i$=class1$|X_i$=$x_i)$; in general, $m-1$ such image planes are needed, where m is the number of classes. Thus, the vector quantities $\mathbf{q}_i^0$ may be represented *in a form identical to the original image data* and presented to the classifier's training phase to generate another probability tree — T^1 say. However this tree must be based on a neighbourhood larger than just the one pixel, otherwise no new information will be considered and so identical PPs will always (in principle at least), be generated. Clearly this process may be repeated to generate a sequence of classifiers T^1, T^2, T^3 etc directly implement the updating functions $\mathbf{g}^1, \mathbf{g}^2, g^3$ etc. The initial tree, T^0 implements the function $\mathbf{D}(Y_i|x_i)$. These classifiers may then be sequentially applied to other similar images as shown on the right-hand side of figure 2.

Convergence

A problem with many PRL updating schemes is knowing when to stop applying them. The problem is significant since classification error rates will invariably deteriorate after a few applications. This stems from the fact that, after the first iteration at least, they loose their theoretical basis [6] and so no longer generate genuine (honest) probabilities. In theory, this is not a problem for the IOU scheme – from its definition, the error rate (over a representative sample) can never get worse, although convergence of the pixel PPs has not yet been formally established.

As has been mentioned however, the definition of IOU functions is a vacuous one and many assumptions and approximations have had to be added before its practical realization as TPR. These fall into two main groups:

1. the training and pruning images are fully representative of the problem domain;
2. the classifier system used accurately determines the underlying distributions.

Neither of these will hold precisely, and the former in particular will become increasingly invalid as the updating proceeds since the images are required to be representative of *joint* distributions of pixels in larger and larger regions.

Never-the-less, it has been found in practice that improvements continue to be made up to about the fourth or fifth update, and that after this any deterioration is only slight; the update functions essentially become weaker in that they make smaller alterations to the assigned PPs.

[3] In fact a larger window can be used at this initial stage to permit the system to see texture or edge/line features.

Experimental results

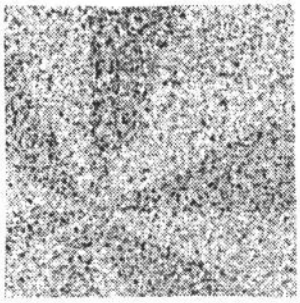
The test image derives from a texture classification problem. Figure 3 shows the initial, non-contextual classification (ie after applying T^0), and the results after 1 and then 4 applications of TPR. The true scene is also shown. Note that the classifiers were trained and validated on similar but not identical images to the ones shown here. The table below shows, for each stage of TPR, the actual and *predicted* error rates. Actual error-rate is obtained by a straightforward comparison between the true scene classification and the MAP decision, based on the generated PPs. Predicted error rate is defined, for the kth iteration, as $\frac{1}{n}\sum_{i=1}^{n} 1 - max_\alpha\ q_{i\alpha}^k .100$ where $q_{i\alpha}^k$ is the αth component of $\mathbf{q}_i^k$. Note that the above definition does not require knowledge of the true scene. It is easy to show that if $\mathbf{q}_i^k$ are genuine (honest) *a posteriori* probability vectors, then the actual and predicted error rates should be (statistically) equivalent. Thus, a comparison of these two quantities gives a useful indication of the extent to which probabilistic classifer is producing honest *a posteriori* probability vectors.

	Initial	1st	2nd	3rd	4th
Actual %error	38.9	13.2	7.5	6.3	6.4
Pred. %error	23.0	11.3	6.9	5.6	5.2

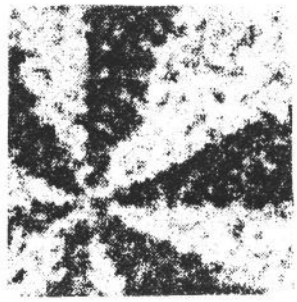
Conclusions

This paper has analysed the performance of a definitively optimal probabilistic relaxation labeling scheme, to produce the following theorems:

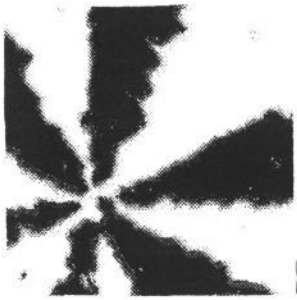
- When pixel data is class-conditionally independent, replacing the original data with the per-pixel PP vectors does not discard any information relevant to subsequent contextual classification. (Theorem 1).
- IOU *cannot* attain a classification that is equivalent to a contextual classification based directly on all the original PPs in the image. (Theorem 2).
- Beyond the first iteration, the IOU functions depend on the *data* distributions, not only on the parameters of the Markov field. This makes it clear



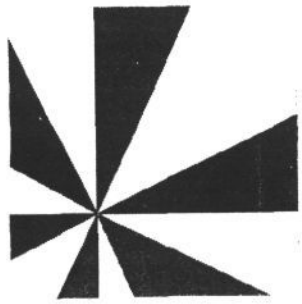
(a) Initial PP image



(b) 1st stage of TPR



(c) 4th stage of TPR



(d) True classification

Figure 3: Results

that any attempt to find an expression for these update-functions would be extremely troublesome. (Theorem 3).

- The IOU function for the second iteration (and almost certainly beyond), is in general, different to the first.

It has been shown the optimal scheme can be implemented by direct estimation of the required distributions, however the technique, known as "trained probabilistic relaxation" (TPR) makes only qualified claims to optimality due to sampling and density estimation difficulties. TPR has been shown to be highly effective in practice.

Acknowledgement

The author is indebted to Paul Otto and Derek Long for the very considerable contribution they have made to the ideas presented in this paper.

References

[1] **L Breiman**, J Friedman, R Olshen et al, *Classification and regression trees,* Wadsworth International Group, 1984.

[2] **A P Dawid**, "Probability forecasting", *Encyclopedia of statistical science,* vol. 7, pp. 210-218, Wiley, 1986.

[3] **P A Devijver** and J Kittler, *Pattern recognition : a statistical approach,* Prentice Hall, 1982.

[4] **D E Goldberg**, *Genetic algorithms in search, optimization and learning,* Addison-Wesley, 1989.

[5] **J Kittler** and J Illingworth, "Relaxation labelling algorithms - a review", *Image and Vision Computing,* vol. 3, no. 4, pp. 206-216, 1985.

[6] **J Kittler** and J Föglein, "On compatibility and support functions in probabilistic relaxation", *CVGIP,* vol. 34, pp. 257-267, 1986.

[7] **S Peleg**, "A new probabilistic relaxation scheme", *IEEE Trans. PAMI,* vol. 2, pp. 362-369, 1980.

[8] **I Poole** and H P Adams, "Lapwing - a trainable image recognition system for the linear array processor", *4th Int. Conf. on Pattern Recognition, BPRA 1988,* pp. 296-305, Springer-Verlag, 1988.

[9] **I Poole**, *Contextual image classification*, Unpublished PhD thesis, submitted 1989.

[10] **G Rayna**, *REDUCE: software for symbolic computation,* Springer-Verlag, 1987.