PARALLEL MODEL BASED SEGMENTATION USING A 3^{rd} ORDER HIDDEN MARKOV MODEL

P. G. Ducksbury

Defence Research Agency, RSRE Malvern, UK.

ABSTRACT.

This work is based upon an algorithm which uses a third order Hidden Markov Mesh Random Field (HMMRF) for the segmentation of images, the algorithm was described initially by Devijver (1), (2). The main emphasis of this paper is to illustrate two important uses of the algorithm. Firstly, how the basic algorithm can be used to segment out urban and non-urban regions from airborne images. Secondly it will be illustrated how the algorithm can be enhanced in order to model textures, the resulting 'texture models' can then be fused together and used to segment images which contain several Brodatz textures.
Finally an apparently basic but very effective method of improving the performance by implementing the approach onto a multi-transputer array is described.

INTRODUCTION.

The fundamental idea behind this work is that given some image, we wish to segment it into a number of homogenous regions. The grouping of the pixels in the image into these regions is based upon local properties and neighbourhood relationships. It is these neighbourhood relationships (*that will be referred to as 'contextual information'*) that are encoded into a set of transitional probabilities in the Markov model.
The basic algorithm [1] has essentially two stages namely the labelling stage and the learning stage. The former is concerned with the modelling of the image (*the segmentation of the image into homogenous regions ie. the assignment of an optimum label to each image pixel*) whilst the latter is concerned with the parameter estimation problem (*ie. taking the current optimum labelling and using it to improve the model parameters*). The algorithm processes an image pixel by pixel in a raster manner and has the advantage in that the learning stage is unsupervised and the only essential parameter required to be entered by the user is the number of states that are required in the model.

DEFINITION OF A 3rd ORDER MMRF.

Firstly, let the image be $M \times N$ in size and then $\mathbf{X}_{M,N}$ denotes a set of feature vectors $X_{m,n}$. In the simplest case these feature vectors will simply be the gray level intensity. Let $\mathbf{\Lambda}_{M,N}$ denote a set of labels $\lambda_{m,n}$, where a label will define the class to which a particular pixel of the image belongs.

$$P(\lambda_{m,n}/\{\lambda_{k,l}|k < m \quad or \quad l < n\}) = P\left(\lambda_{m,n} \Big/ \begin{array}{l} \lambda_{m-1,n-1} \ \lambda_{m-1,n} \\ \lambda_{m,n-1} \end{array} \right)$$

for all points (m,n) such that $1 < m \leq M$ and $1 < n \leq N$, with boundary conditions existing along the first row and column.
The MMRF is said to be 3rd order due to the fact that $\lambda_{m,n}$ has a dependency [2] upon its three neighbouring labels, namely $\lambda_{m-1,n-1}$, $\lambda_{m-1,n}$ and $\lambda_{m,n-1}$.
The model is assumed throughout to be *spatially homogeneous*, meaning that we can assume that the transitional probabilities are totally independent from the position in the image. This now allows us to write $P_{q/rst}$ as an abbreviation of $P(\lambda_{m,n} = q/\lambda_{m-1,n} = r; \lambda_{m-1,n-1} = s; \lambda_{m,n-1} = t)$ where q, r, s and t are contained within the state-space of the model. This is read as the probability of label q being chosen given the three neighbouring labels r, s and t.

THE ALGORITHM.

In the initial stages of the algorithm a model is created by fitting a set of Gaussian distributions (*one per state, denoted by* $p_q(X)$) to the histogram of the image. The model also consists of an initial set of transitional probabilities which are defined under the assumption that any two neighbouring pixels in the image are more likely to have values close to each other than at either end of the gray level spectrum. The labelling stage is then a 2-dimensional set of recurrence relations described in (1) and (2) which uses the information contained in $p_q(X)$ together with the set of transitional probabilities $P_{q/r,s,t}$ in order to cluster the image pixels into a set of homogenous regions.

---

[1] *limited space prevents a complete description of the mathematics of both the algorithm and its derivation, for those interested refer to (1) and (2).*

[2] *In a second order MMRF the dependency of $\lambda_{m,n}$ would exist with the two neighbours $\lambda_{m-1,n}$ and $\lambda_{m,n-1}$.*

To appear
'IEE 4^{th} Int. Conf. Image Processing and its applications', Maastricht, The Netherlands, 7-9 April 1992.

The learning stage is based upon a class of techniques which are known as *decision directed.* The decision directed re-estimation technique effectively transforms the updating formula into a set of relatively simple counting formula, where I[:] is taken to be an indicator function[3]. The learning stage can now be described as follows, the $P_{q/r,s,t}$ is given by the following expression.

$$\frac{\sum_{m=2}^{M}\sum_{n=2}^{N}\mathbf{I}\begin{bmatrix} \tilde{\lambda}_{m-1,n-1}=s & \tilde{\lambda}_{m-1,n}=r \\ \tilde{\lambda}_{m,n-1}=t & \tilde{\lambda}_{m,n}=q \end{bmatrix}}{\sum_{m=2}^{M}\sum_{n=2}^{N}\mathbf{I}\begin{bmatrix} \tilde{\lambda}_{m-1,n-1}=s & \tilde{\lambda}_{m-1,n}=r \\ \tilde{\lambda}_{m,n-1}=t & \end{bmatrix}}$$

with special cases existing for the first row and column. For instance the first column is defined as

$$P_{q/r} = \frac{\sum_{n=2}^{N}\mathbf{I}[\tilde{\lambda}_{n-1,1}=r, \tilde{\lambda}_{n,1}=q]}{\sum_{n=2}^{N}\mathbf{I}[\tilde{\lambda}_{n-1,1}=r]}$$

and similarly for $P_{q/t}$ along the first row. The set of distributions p_q are given by

$$p_q(\zeta_i) = \frac{\sum_{m,n|x_{m,n}=\zeta_i}\mathbf{I}[\tilde{\lambda}_{m,n}=q]}{\sum_{m,n}\mathbf{I}[\tilde{\lambda}_{m,n}=q]}.$$

The above equations are performed for all combinations of q, r, s and t in the state space. Essentially therefore it is simply examining every possible 2×2 neighbourhood throughout the image and counting the number of occurences of particular combinations (*or patterns*) of labels.

SEGMENTATION OF URBAN REGIONS.

In this instance the images that we are dealing with are infra-red linescan images taken over the Bedfordshire area at a height of 3000 feet. We wished to look at techniques for the automatic location of features such as urban regions (*also, but not mentioned here, the location of road networks*). Because of certain constraints such as being limited to single band 8-bit data the approach taken is as follows. The image is preprocessed to obtain some statistical information which can be used in an attempt to classify its content into urban/non-urban. It will be seen that this statistical classification is noisy and rather basic, the HMMRF algorithm is then used to 'tidy up' this classification by segmenting it into homogenous regions.
A mesh of windows is placed over the image, with each window being typically 16×16 pixels in size. For each window a set of elementary statistics are computed.

- the number of significant edges.
- the number of significant extrema.
- comparison of the histogram using a chi-squared measure with a set of standard Gaussians.

These are presented in order of their effectiveness and are coded into a single statistic per window. Figure 1 illustrates the computed statistics for an image whilst figure 2 shows the original image with an outline of the urban region indicated (*outline extracted using a Sobel operator on the segmented image*). The results are interpreted as follows, for a simple 2-state model we have only urban or non-urban regions, for any model with more than 2 states we have regions of high probability of being urban through to regions of low probability of being urban. The result shown in figure 2 is for a 4-state model with a single iteration, here only the region that has the highest probability of being urban is illustrated[4].
The window size can be reduced to give a finer outline of the urban region and a number of iterations can be used to allow the learning stage to improve the initial estimate. In practice it has been found that although the window size can be reduced to 6×6 pixels the best results do occur at around 16×16. This is partly due to the fact that as the window size is reduced the statistics being used become less reliable and in addition to this the number of learning iterations required to remove errors in the classification increases. One of our main objectives is for a coarse level processing of an image in the shortest possible time.

SEGMENTATION OF BRODATZ TEXTURES.

For the segmentation of Brodatz textures (3) the basic idea is that if it is possible to derive a model [5] which describes a texture then it should also be possible to fuse several of these models together. We should then be able to segment a composite texture image using the new composite model that has been obtained.

Combining Markov Models.

The process of combining several models together itself makes use of a model which describes the rules for the merge. Figure 3 illustrates this procedure for the merging of two 2-state models into one 4-state model. The notation $m_{i,j}^{3}$ denotes the probability of making a transition from model i to model j. Whereas previously for a two state model the process could either remain in the current state or make a transition into its neighbouring state, now however a transition can also be made into either of the states that are in the second model.
Each texture is histogram equalised independently

[3] *If the expression within the brackets is true then the indicator function returns a 1 otherwise it returns a 0.*

[4] *These regions could be considered as a set of contours .*
[5] *the use of the word derive in this sense means to train on a given texture using the learning stage of the algorithm.*

prior to the start of the process to remove the possibility of detection due to first order statistics. The method has the property of rapid convergence to a local maximum and for all of these tests the algorithm was run with a 2-state model for 10 learning iterations to obtain the set of model parameters p_q, P_q, $P_{q/r}$, $P_{q/t}$ and $P_{q/r,s,t}$ for each texture. Once these have been obtained the models will then be merged into a composite model and this can be applied to a composite image of the two textures.
The transitional probabilities used in model 3 allowed for a 90% probability of staying in the current texture model and a 10% probability of making a transition into another texture model.
Figure 4 shows a 4-state segmentation of a composite texture (*2 states per texture*), first with the contextual information from the transitional probabilities $P_{q/r,s,t}$ removed and then with the contextual information included. (*the results are displayed such that if a given pixel has a label with a value corresponding to either of those in model 1 then it is displayed as black , otherwise it is displayed as white*).
The segmentation without contextual information is basically the segmentation using knowledge of the differences in the histograms of the two textures. Whilst the segmentation with contextual information is where the Markov Model is used to resolve the ambiguities.

PARALLEL IMPLEMENTATION.

The algorithm as it appears in the literature is inherently sequential and hence could impose problems from the point of view of processor time. However a version has been experimented with which is based upon geometric parallelism, here the problem is divided up into smaller ones which can then be run in parallel.
The image is split into a number of equally sized subimages each being allocated to a particular processor, obviously row 1 of subimage n is now treated as the first row of an image instead of the n^{th} row, thus we might naturally expect the resulting segmentation to be of a lower quality [6]. In practice the modelling of the distributions $p_q(X)$ is done sequentially on the complete image prior to parallelising, hence each processor has knowledge of the global distribution of the image and not just its particular subimage, (*if this is not done then the quality of the segmentation will be unacceptable, it will tend to have a very banded structure corresponding to the divisions over the processor boundaries.*)
Table 1 shows the approximate processor times required in seconds. Here the sequential case is on a VaxStation 3200, whilst the parallel is on 16 T800 transputers. It is important to realise that the urban region location takes as input a small set of statistics instead of a large gray level image.
The statistics are not yet computed in parallel, however it is likely to be a trivial task as each window can be computed in parallel (*no boundary communication is required*). A processor farm might be appropriate in this instance.

[6] *This approach is not strictly algorithmically valid but is being considered because of its obvious advantages.*

DISCUSSION.

This work has illustrated that the basic algorithm proposed by Devijver can be used and readily extended for a number of cases; firstly, as a standard (*contextual*) segmentation tool used on preprocessed data for the location of urban and non-urban regions in airborne images, secondly, for the development and fusion of texture models which can then be used for the segmentation of images composed of several textures. (*Previous work (5) illustrated how this approach could be used for the location of driveable regions for an autonomous land vehicle.*)
It has also been shown how the performance can be greatly improved through the use of parallel architectures. At present the quality of the segmentation appears comparable with those from the standard sequential case, but the approach will need to be examined in more detail with reference to this and also other possible architectures and approaches for parallelism.

ACKNOWLEDGEMENTS.

This work was partially funded by the UK, DTI Information Engineering Programme : Project IED-1936 *'Vision by Associative Reasoning'*.

REFERENCES

1. Devijver, P.A., Oct 1986, *"Segmentation of binary images using 3rd order Markov Mesh image models"*, Proc. 8th Int. Conf. Pattern Recognition, Paris, pp 259 - 261.

2. Devijver, P.A., 1988, *"Image segmentation using Causal Markov Random Field models"*. Pattern Recognition. J.Kittler (ed) Lecture Notes in Computer Science 301, Springer-Verlag.

3. Brodatz, P., 1966, *"Textures - a photographic album for artists and designers"*. Dover, New York.

4. Ducksbury, P.G., 1990, *"Experience in image processing on a transputer using a third order Hidden Markov Mesh Random Field "*. RSRE, SP4 Research note 117.

5. Ducksbury, P.G., 1991, *"Model based texture segmentation using a 3rd order Hidden Markov Model"*, RSRE SP4 Research Note 143.

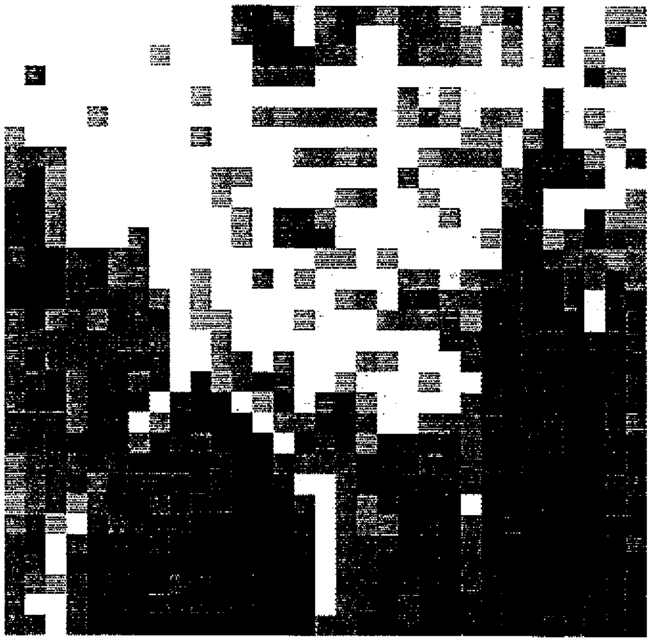

Figure 1: *Window Statistics.*

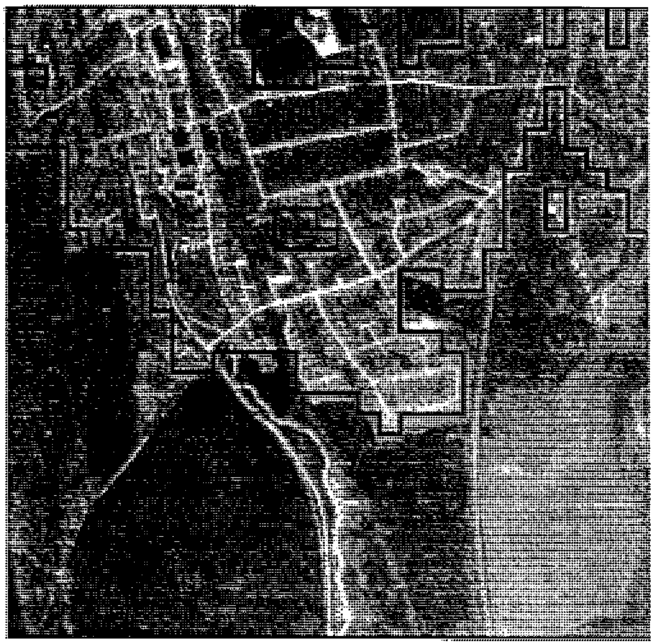

Figure 2: *Located Urban Region.*

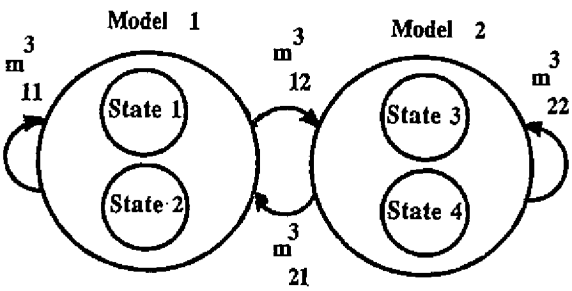

Figure 3: *Combining Markov Models.*

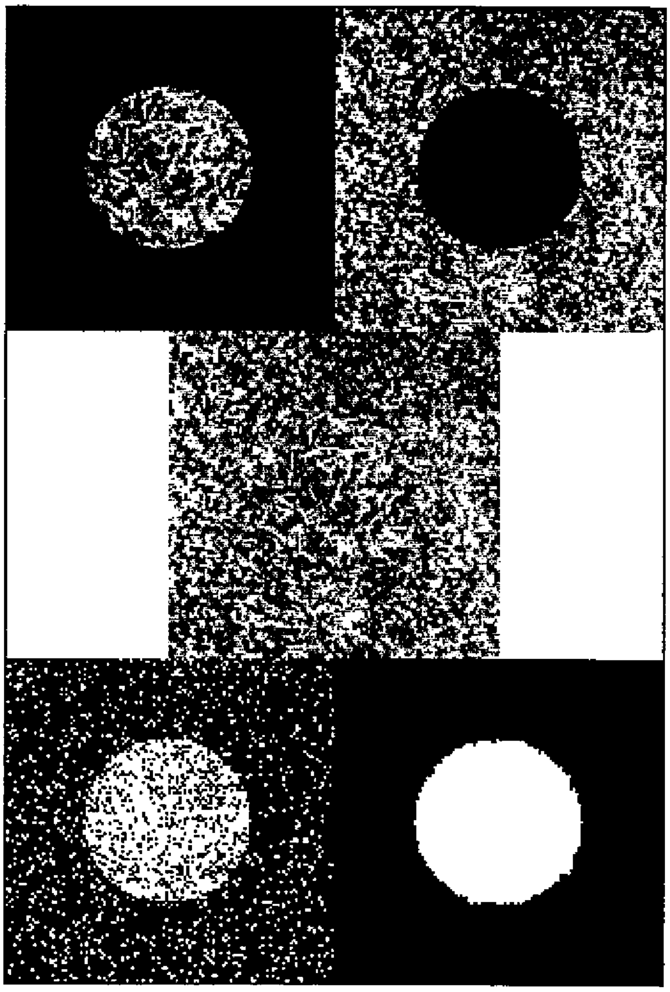

Figure 4: *Segmentation. (Top row : Use of masks to extract regions from two Brodatz textures, Middle row : the composite image, Bottom left : segmentation without context, Bottom right : segmentation with context)*

Problem	*Size*	*Seq*	*Par*
Urban	496x496		
-Statistics	31x31	30	n/a
-HMMRF	31x31	30	2
Brodatz	128x128	180	14

Table 1: *Processor Performance*