

Parallel Texture Region Segmentation using a Pearl Bayes Network

P.G.Ducksbury
Defence Research Agency, Malvern, UK.

Abstract

This paper describes a parallel implementation of a texture segmentation algorithm. The algorithm uses a Pearl Bayes Network (PBN) to combine evidence for the location of urban regions in airborne infra-red linescan images and for the location of driveable regions in autonomous land vehicle imagery. A multilevel PBN approach is introduced and followed by an example which is used to illustrate the derivation of the propagation and fusion equations. The parallel implementation is then described with results demonstrating its effectiveness.

1 Introduction

The vast majority of papers in the literature which deal with the fusion of knowledge for applications in evidential reasoning have concentrated on the theoretical aspects. The papers have dealt with the structure of the networks themselves and how they can be used to represent and manipulate knowledge. The Bayesian approach for reasoning is described in a series of papers by Pearl in particular [4] where he describes the basics of Bayesian networks and belief functions. There are a number papers which are now beginning to address the problems of using evidential reasoning in the area of image understanding. Some of these are reviewed in [2] and this paper describes one such practical application.

2 The Problem

The problem is generically defined as the location of some region in an image. This problem will be approached by taking several statistical measures from small patches of an image these are treated as a set of judgements (*virtual evidence in Pearl's notation*) about the content of the patches. This evidence is then combined into a belief of the patch belonging to the defined region.

Ideally we would like to be able to improve the estimate (*or belief*) and in some way this can be achieved by using information from a higher level. An example of this being given in figure 1 which shows one way in which a multilevel approach can be applied. The lowest level in the pyramid contains the raw pixels at full image resolution, whilst the following level contains the raw pixels at half

resolution. The next two levels contain statistics computed from the image data. At the top level is the overall belief in a region.

S_1 - S_3 : Edges, Extrema, Distribution type from fine resolution image.

S_4 - S_6 : Edges, Extrema, Distribution type from coarse resolution image.

B_f : Belief from fine resolution.

B_c : Belief from coarse resolution.

B : Belief

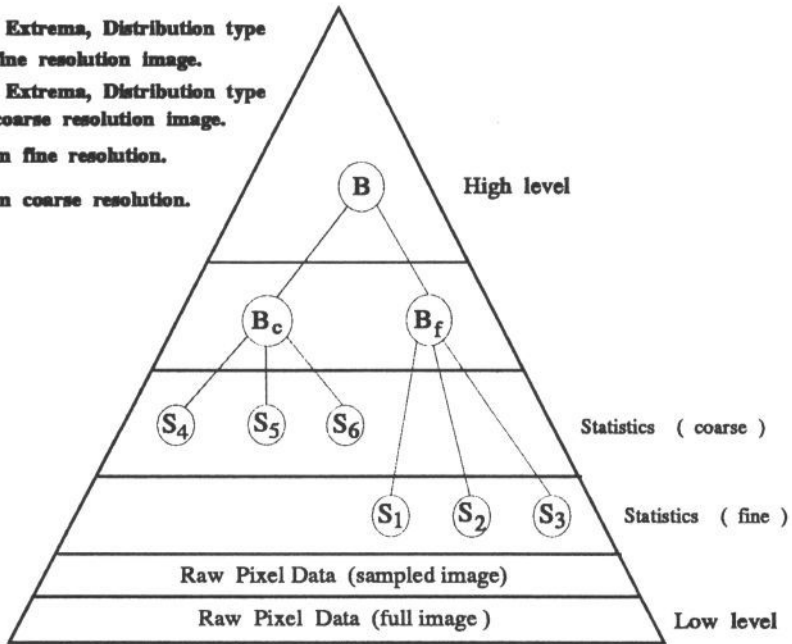


Figure 1: Multi-level Pyramid.

In the next section we will concentrate on the simpler case of just a single level in the network as shown in more detail in figure 2. It will be shown how the equations for the belief and propagation of information in a PBN can be derived for a given example.

3 Network and equation construction

Definition 3.1 *Bayesian networks are directed acyclic graphs, such that a graph G is a pair of sets (V,A) for which V is non-empty. The elements of V are vertices (nodes) and the elements of A are pairs (x,y) called arcs (links) with $x \in V$ and $y \in V$.*

Considering the comparatively simple network shown in figure 2 the equations for computing the belief and propagation of information are slightly more complicated than they would at first appear. In the simplest case of a causal tree a given node would have just one predecessor, in our network the node A has several predecessors. The equations are derived along similar lines to those derived by Pearl in [4], where in his example node A has just two predecessors and two successors.

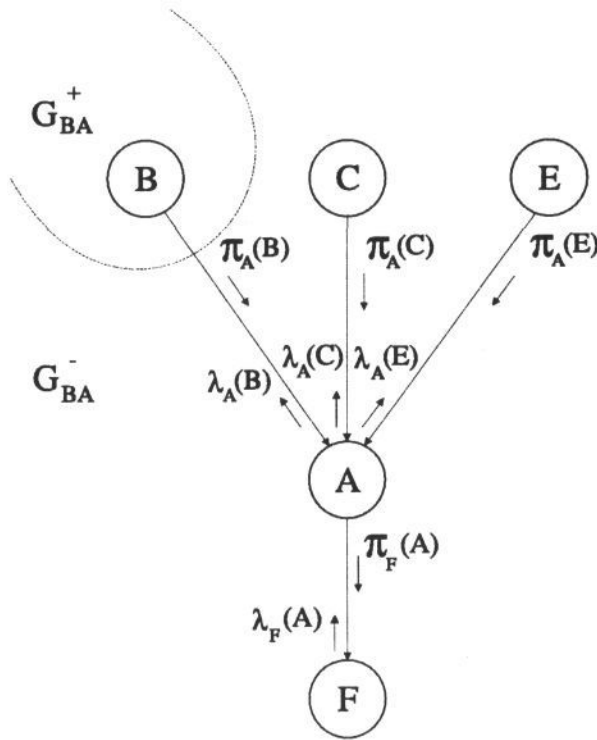


Figure 2: Pearl Bayes Network : Nodes B, C and E will represent different statistical information, Whilst node A represents our belief in an Urban patch.

If we consider the link from node B to A then the graph G consists of the two subgraphs G_{BA}^+ and G_{BA}^- . These two subgraphs contain the datasets D_{BA}^+ and D_{BA}^- respectively.

3.1 Belief equations

From figure 2 we can see that node A separates the two subgraphs $G_{BA}^+ \cup G_{CA}^+ \cup G_{EA}^+$ and G_{AF}^- . Given this fact we can write the equation :

$$P(D_{AF}^- | A_i, D_{BA}^+, D_{CA}^+, D_{EA}^+) = P(D_{AF}^- | A_i) \quad (1)$$

by using Bayes rule the belief in A_i can be written as

$$\begin{aligned} BEL(A_i) &= P(A_i | D_{BA}^+, D_{CA}^+, D_{EA}^+, D_{AF}^-) \\ &= \alpha P(A_i | D_{BA}^+, D_{CA}^+, D_{EA}^+) \cdot P(D_{AF}^- | A_i) \\ &= \alpha P(D_{AF}^- | A_i) \cdot \left[\sum_{j,k,l} P(A_i | B_j, C_k, E_l) \cdot P(B_j | D_{BA}^+) \cdot P(C_k | D_{CA}^+) \cdot P(E_l | D_{EA}^+) \right] \end{aligned} \quad (2)$$

where α is taken to be a normalizing constant. It can be seen that equation 2 is computed using three types of information

- Causal support π (from the incoming links).
- Diagnostic support λ (from the outgoing links).
- A fixed conditional probability matrix (which relates A with its immediate causes B, C and E).

The equations which form the above information are given as follows. Firstly the causal support equations :

$$\pi_A(B_j) = P(B_j|D_{BA}^+) \quad (3)$$

$$\pi_A(C_k) = P(C_k|D_{CA}^+) \quad (4)$$

$$\pi_A(E_l) = P(E_l|D_{EA}^+) \quad (5)$$

Secondly the diagnostic support equation is given by

$$\lambda_F(A_i) = P(D_{AF}^-|A_i) \quad (6)$$

Finally the conditional probability matrix is defined to be

$$P(A|B, C, E) \quad (7)$$

Equation 2 can now be rewritten in order to obtain the belief at node A .

$$BEL(A_i) = \alpha \lambda_F(A_i) \cdot \sum_{j,k,l} P(A_i|B_j, C_k, E_l) \cdot \pi_A(B_j) \cdot \pi_A(C_k) \cdot \pi_A(E_l) \quad (8)$$

The belief at nodes B, C and E can be obtained from the equations

$$BEL(B_j) = \alpha \cdot \pi_A(B_j) \cdot \lambda_A(B_j) \quad (9)$$

$$BEL(C_k) = \alpha \cdot \pi_A(C_k) \cdot \lambda_A(C_k) \quad (10)$$

$$BEL(E_l) = \alpha \cdot \pi_A(E_l) \cdot \lambda_A(E_l) \quad (11)$$

3.2 Propagation equations

The propagation equations for the network are derived as follows, firstly the diagnostic ones. From an analogy with equation 6 we can write

$$\lambda_A(B_i) = P(D_{BA}^-|B_i) \quad (12)$$

by partitioning the D_{BA}^- into its component parts, namely A , D_{AF}^- , D_{CA}^+ , D_{EA}^+ we can obtain

$$\lambda_A(B_i) = \alpha \sum_{j,k} \left[\pi_A(C_j) \cdot \pi_A(E_k) \cdot \sum_l \lambda_F(A_l) \cdot P(A_l|B_i, C_j, E_k) \right] \quad (13)$$

likewise for $\lambda_A(C_j)$ and $\lambda_A(E_k)$

$$\lambda_A(C_j) = \alpha \sum_{i,k} \left[\pi_A(B_i) \cdot \pi_A(E_k) \cdot \sum_l \lambda_F(A_l) \cdot P(A_l|B_i, C_j, E_k) \right] \quad (14)$$

and

$$\lambda_A(E_k) = \alpha \sum_{i,j} \left[\pi_A(B_i) \cdot \pi_A(C_j) \cdot \sum_l \lambda_F(A_l) \cdot P(A_l|B_i, C_j, E_k) \right] \quad (15)$$

The Causal equations are defined as follows. From an analogy with equation 3 we can write

$$\pi_F(A_i) = P(A_i|D_{BA}^+, D_{CA}^+, D_{EA}^+) \quad (16)$$

and then derive the equation

$$\pi_F(A_i) = \alpha \left[\sum_{j,k,l} P(A_i|B_j, C_k, E_l) \cdot \pi_A(B_j) \cdot \pi_A(C_k) \cdot \pi_A(E_l) \right] \quad (17)$$

An important point to realise is the fact that equations 13 - 15 and equation 17 demonstrate that the parameters λ and π are orthogonal to each other ie. perturbation of one will not affect the other. Hence evidence propagates through a network and there is therefore no reflection at boundaries.

4 Implementation

The technique is demonstrated for two different sets of imagery, firstly the segmentation of urban from non-urban regions in infra-red linescan images and secondly, the detection of driveable regions in a sequence of forward looking images from a moving vehicle. For both sets the technique is as follows :-

A Mesh is placed over an image and for each of the resulting windows a set of statistics (*which provide strong texture discrimination*) are computed. The statistics are the number of edges, the number of extrema and gray level distribution type. These statistics are quantized down into a smaller number of levels. The number of edges and extrema are both reduced to 5 levels, whilst the distribution type remains with its 4 possibilities.

The statistics are then used to produce a set of judgements, for example an expert might upon looking at a particular window issue a report of the form (0.0, 0.7, 0.9, 0.6, 0.0). This means that he believes there is a 70% chance that level 2 describes the number of edges, 90% chance that its level 3 and 60% for level 4. But he believes there to be no chance of it being levels 1 or 5.

For the Belief at nodes B_f , B_c and B in figure 1 it was decided to have 3 variables which denote the possible values that the region can have, namely (low, medium, high).

The fixed conditional probability matrices (eg $P(B_f|s_1, s_2, s_3)$ etc) which are the prior information and relate the given node with its causal information are

created along similar lines to that used in [2] and which originally came from [1]. They are based upon the assumption that the probability of an event at a given node should be greater if its causal information is tightly clustered together than it should be if the causal information is further apart. For the $P(B|Bf, Bc)$ matrix (which relates the beliefs from the fine and coarse resolutions) slightly more emphasis is given to the causal information received from the coarse resolution belief.

$P(Bf_i|s1_j, s2_k, s3_l)$ is described formally as

$$P(Bf_i|s1_j, s2_k, s3_l) = \begin{cases} 0.75 & \text{if } i = j = k = l \\ 0.25/\alpha & \text{if } (i \neq j = k = l) \wedge (0 < |i - j| \leq C) \\ 1.0/\beta & \text{if } \neg (j = k = l) \\ & \wedge (\max(j, k, l) - \min(j, k, l) \leq 2C) \\ & \wedge (\min(j, k, l) \leq i \leq \max(j, k, l)) \\ 0.0 & \text{otherwise} \end{cases} \quad (18)$$

such that $\sum_{j,k,l} P(Bf_i|s1_j, s2_k, s3_l) \leq 1 \quad \forall i$

where

$$C = 1,$$

i, j, k, l range over the number of variables in Bf , $s1$, $s2$ and $s3$ respectively.

α and β represent the number of different values of i satisfying the constraint.

$P(B|Bf, Bc)$ is defined as

$$P(B_i|Bf_j Bc_k) = \begin{cases} 0.9 & \text{if } i = j = k \\ 0.7 & \text{if } (i = j) \wedge (|i - k| \leq 1) \\ 0.3 & \text{if } (i = k) \wedge (|i - j| \leq 1) \\ 0.6 & \text{if } (i = j) \wedge (|i - k| > 1) \\ 0.1 & \text{if } (i = k) \wedge (|i - j| > 1) \\ 0.0 & \text{otherwise} \end{cases} \quad (19)$$

such that $\sum_{j,k} P(B_i|Bf_j Bc_k) \leq 1 \quad \forall i$

where

i, j, k range over the number of variables in B , Bf and Bc respectively.

If the application remains of a broadly similar nature (ie classifying (or clustering) regions) then the only change necessary would perhaps be a new set of statistics which more accurately describe the detail required in the image. In addition to this if the number of input nodes alters then the prior knowledge in the fixed conditional probability matrix will need to change, however the set of basic equations given above can be used to automatically generate this information.

5 Parallel Processing

5.1 Architecture

The architecture used for this work is based upon a transputer array and is called CHIP (Conceptual Hierarchical Image Processor). It is a real-time image processing system which is intended to be used as a test-bed for developing new algorithms, and prototyping of new image processing architectures.

To avoid a communication bottleneck video data is communicated throughout using two digital video busses. CHIP provides for video input to an acquisition/display board which is connected to a video crossbar switch, this in turn is connected to the main processing unit of CHIP. Various DSP devices can be flexibly interconnected by the video crossbar for the pre-processing of images. The result can then be passed to all processors in the transputer array.

The processing array currently consists of 20 modules each being a T805 transputer with 4MByte of DRAM and 2MByte of VRAM. The latter forming two banks of 4 (512×512) framestores. Two video input and output busses are common to each module, each bank of framestores being connected to one of the input and one of the output busses. The individual transputer links of each module are connected to a link crossbar switch allowing for different network topologies. Details of the high level system design for CHIP can be found in [5].

5.2 Algorithm

In considering the implementation there are probably a number of ways in which the algorithm could actually be parallelised. Given below are just a few options that were considered for exploiting the parallelism.

1. Topological configuration of CHIP to match the Multilevel PBN tree structure.
2. Algorithmic Parallelism. Using multiple pipelines in which the first processor in each computes the statistical evidence, whilst the next processor computes the pbn belief.
3. Geometric Parallelism. Each processor handles a small section of the data space and performs all the computation of the algorithm.

The geometric approach was the one chosen as the most suitable it consists basically of a complete copy of the algorithm (statistics and PBN) on each slave processor with one processor allocated as a master controller, this corresponds to a processor farm approach. This approach was chosen as all stages of the algorithm are totally deterministic and since once the information regarding the PBN tree structure and probabilities has been communicated to each processor and the tree built no further communication between processors is required. This eliminates the need for any (possibly) expensive communications and opens the way for expectation of significant speedups as the number of processors is increased.

The physical processors are arranged in a bidirectional pipeline as shown in figure 3. Currently the master reads an image file from the host computers disc and sends it to the acquisition and display (ACQDISP) controller which sets up the the framestore control registers to display the image and then sends it to each slave processor via the video crossbar switch. The controller then sets up a map specifying which processors are responsible for which windows of the image. This together with the control registers enable the results of each processor to be displayed directly the moment they become available. The bidirectionality of the pipeline is used simply to allow results to be sent back across channels to be saved

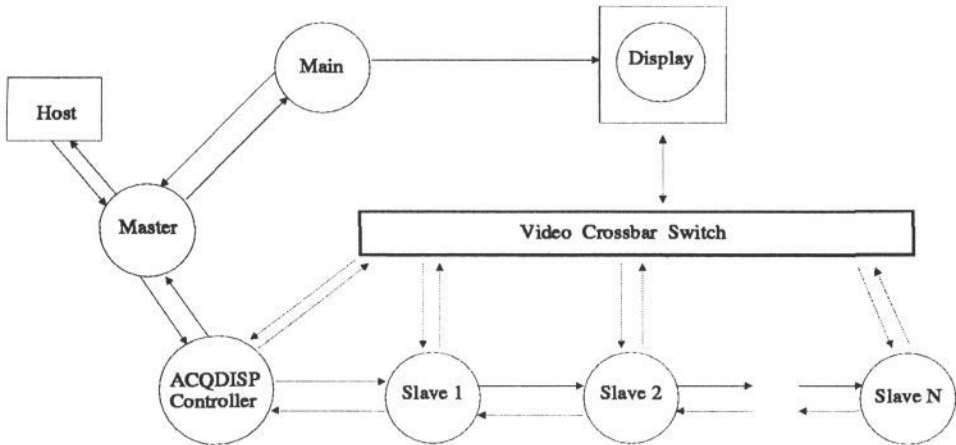


Figure 3: Hardware Process Structure.

to a file. These results are also used by the master process to send instructions to the 'Main' process which is used to produce overlays and graphics on the display.

The load balancing is performed by the master process which works out how many windows there are to be processed, the image being treated as a 1-dimensional set of windows numbered consecutively with the top left being 1. Since the master knows how many slave processes there are in the configuration it can work out the number of windows per processor. Both of these numbers are sent out to the first slave in the pipeline which receives them and subtracts the windows per processor from the total. This slave then sends out to the next slave the number of remaining windows and the number of windows per processor. This process continues until the slave at the end of the pipeline receives the message and handles whatever number of windows are remaining.

6 Results and Conclusions

The results for locating urban regions are given for the image shown in figure 4. An example of the statistics for this image are given in figure 5 using a window size of 16×16 pixels. These statistics are converted into a set of judgements as described in the previous section and then presented as evidence to the PBN. The result for the Multi-level PBN is shown in figure 6. This was produced by drawing an outline of the belief variable which corresponds to the area of highest belief in urban and overlaying onto the original image.

Figure 7 shows the result of the multi-level PBN on a driveable region image using the same three input nodes at each level, representing extrema, edges and distribution type respectively. The white patches represent the centre of windows that have been classified as driveable.

For the both cases the time required with $20 \times T805$ transputers was 0.95 seconds given a 500×500 image. These times are the computational time required to compute all the statistical measures and run the multi level PBN for all windows.

It has been demonstrated that the approach gives good results on two different classes of application imagery and that once initialisation has been performed the algorithm is ideally suitable for asynchronous parallel implementation for which a linear speedup obtainable. Realistically the degree of parallelism is only limited by the number of processors that are available, each of the windows could theoretically have been computed in parallel. This approach compares extremely favourably with a previous one described in [3] which used a Hidden Markov Mesh Random Field for the texture region segmentation of the statistical data, and was not as suitable for a parallel implementation.

The approach has been tested on a variety of images at several different altitudes with considerable success. The approach also performs well for the alv imagery and has been used with a live video as the input with the algorithm grabbing input frames as fast as it can. It may in future be possible to perform some of the initialisation and statistical computations in hardware. It is estimated that with some of the initialisation performed in hardware and an array of T9000 transputers the driveable region version should be able to perform at the rate of at least 12-15 frames per second.

References

- [1] Devijver P.A., '*Real-time modeling of image sequences : Based on Hidden Markov Mesh Random Field Models*' in '*Decision making in context*' a Course on Statistical Pattern Recognition by P.A.Devijver and J.Kittler, Surrey University, 1989.
- [2] Ducksbury P.G., '*Evidential Reasoning - A Review and Demonstration.*', UK DTI IED project 1936 : '*Vision by Associative Reasoning*', Report no. VAR-TR-RSRE-92-4, July 1992.
- [3] Ducksbury P.G., '*Parallel Model-Based Segmentation using a 3rd order Hidden Markov Model*', IEE 4 Int. Conf. Image Processing and its Applications, Maastricht, The Netherlands, 7-9 April, 1992.
- [4] Pearl J., '*Fusion, Propagation, and structuring in Belief Networks*', Artificial Intelligence 29 (1986) pp 241 - 288.
- [5] Smith R.W., '*Conceptual Hierarchical Image Processor (CHIP) : System Design*', Issue 1.0, October 1992, CSE1 Division, DRA Malvern.



Figure 4: Raw Image, 3000 ft.



Figure 6: Urban outline.

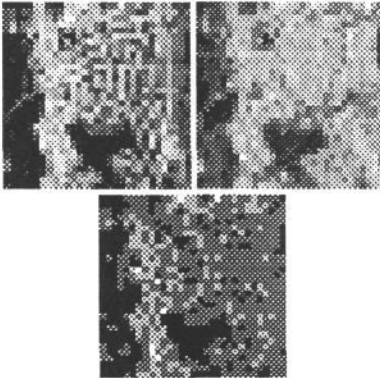


Figure 5: Statistics : Top Left - Edges, Top right - Extrema, Bottom - Distribution type.

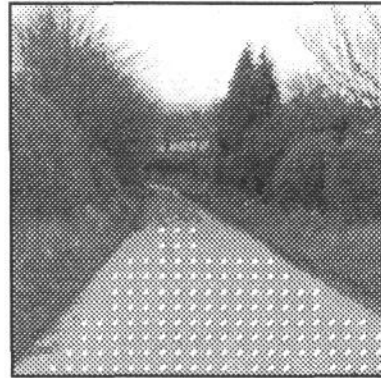


Figure 7: Driveable region

Acknowledgments

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

British Crown Copyright 1993 / DRA
 Published with the permission of the
 Controller of Her Britannic Majesty's
 Stationary Office.