Tina Memo No. 1992-001

'Combining the Opinions of Several Early Vision Modules Using a Multi-Layered Perceptron.'

Int.Journal of Neural Networks,2,2/3/4,June-December,75-79,1991. some figures missing.

see also: 'Data Fusion Using an MLP'.

Proc. British Machine Vision Conference, Glasgow,339,1991.

Combining the Opinions of Several Early Vision Modules using a Multi-Layer Perceptron.

D M Booth, N A Thacker, J E W Mayhew and M K Pidcock

Last updated
1 / 09 / 2005

This document forms part of the **Features and Measurement** Series available from www.tina-vision.net.

2002-005	Tutorial: An Empirical Design Methodology for the Construction of Machine Vision Systems.
1996-001	Algorithmic Modelling for Performance Evaluation.
2006-002	A Statistical Framework for Detection of Connected Features.
2006-007	Quantitative Verification of Projected Views Using a Power Law Model of Feature Detection.
1997-003	Tutorial: Supervised Neural Networks in Machine Vision.
1995-003	Invariance Network Architecture.
1992-001	Combining the Opinions of Several Early Vision Modules using a Multi-Layer Perceptron.
2005-006	Curve Fitting and Image Potentials: A Unification within the Likelihood Framework.
1996-002	Tutorial: The Likelihood Interpretation of the Kalman Filter.
1994-003	Using a Switchable Model Kalman Filter.
2004-012	Tutorial: Computing 2D and 3D Optical Flow.
2005-011	Comparing the Performance of Least-Squares Estimators: when is GTLS Better than LS?
1994-001	Tutorial: Overview of Stereo Matching Research.
1992-002	Online Stereo Camera Calibration.
1995-002	Calibrating a 4 DOF Stereo Head.
2000-009	An Evaluation of the Performance of RANSAC Algorithms for Stereo Camera Calibration.
2001-011	The Evolution of the TINA Stereo Vision Sub-System.
2007-011	A Methodology for Constructing View-Dependent Wireframe Models.

WWW.TINA-VISION.NET

Imaging Science and Biomedical Engineering Division,
Medical School, University of Manchester,
Stopford Building, Oxford Road,
Manchester, M13 9PT.

Combining the Opinions of Several Early Vision Modules using a Multi-Layer Perceptron.

D M Booth [1], N A Thacker[†], J E W Mayhew [2] and M K Pidcock [3]

Abstract

This paper deals with the solution of a binary classification problem by acting on the combined evidence of several early vision modules. Each module provides an opinion on the identity of an individual image element based on a specific area of expertise, such as texture, motion, depth etc. The problems involved in reaching a consensus of opinion are discussed, and the effectiveness of using a trained Multi-Layer Perceptron as a tool for data fusion is examined. Some preliminary results are reported. The original version of the document has been extended with new Appendices to cover recent developments in the field of pattern recognition and machine learning. We hope to show with this additional material that this approach to solving the data fusion problem is just as valid now as when we first suggested it, and does not have the failings of more recent and quite widespread approaches.

[1]Research Initiative in Pattern Recognition, DRA Electronics Division, St. Andrews Road, Malvern, Worcs.
[2]AI Vision Research Unit, University of Sheffield, Sheffield, Yorks.
[3]Computing and Maths Dept., Oxford Polytechnic, Oxford.

Introduction

This paper describes the findings of a preliminary investigation into combining evidence generated by several vision modules for the solution of a binary classification problem. More specifically, there is a requirement to combine evidence based on texture (Booth and Mayhew, 1989 & 1990a) and stereo disparity (Booth and Mayhew, 1990b) for segmenting a sequence of binocular stereo images into road and non-road classes. Problems of this type are common in computer vision when there is a desire to recover a three dimensional representation of a scene from its projection onto a two dimensional image plane. Numerous types of vision module have been devised for interpreting noise corrupted features such as shadows, stereo disparities, texture, and optical flow. However, given only the characteristics of the imaging device and the imagery itself, each problem is ill-posed. As a result, early vision modules constrain the possible solutions by incorporating prior knowledge. However, since this may vary for different modules, the problem arises of how to combine their results in a coherent manner (Appendix A). The choice of fusion mechanism requires consideration of the following issues :

• Expert Reliability

The reliability of experts may differ widely. An unreliable expert must be recognised and its opinions given reduced weighting.

• Correlated Opinions

The opinions of experts may be correlated. In such cases, experts who tend to draw the same conclusions may dominate the consensus, though the amount of independent information they represent may not be great. Again, these must be recognised and their opinions weighted accordingly.

• Input Representation

A particular fusion technique may place constraints on the way in which opinions can be expressed. Some approaches require the posterior probability of a particular event, and the goal is to produce a single probability distribution which summarises the various estimates. A survey of such techniques is given by Berenstein et al (1985). In our application, the texture discriminator generates class conditional posterior probabilities. However, the obstacle detector generates unnormalised probabilities which represent a measure of deviation from the ground plane. The fusion technique must be able to handle this. In addition, it would be desirable to fuse opinions, or measures of belief, originating from distributions of a more arbitrary nature.

• Output Representation

A consensus opinion should ideally be in the form of a probability. These are readily converted into class labels, and a set of opinions relating to a local neighbourhood is suitable for input into a probabilistic relaxation scheme for incorporating contextual knowledge.

Here the Multi-Layer Perceptron (MLP) is proposed as an ideal tool for data fusion. It will be shown to satisfy all of the constraints listed above, and in doing so, demonstrates a general purpose facility that many other fusion techniques do not possess. In particular the approach contrasts directly with the method of **Boosting** which has received much attention since the publication of this document, the possible pitfalls with boosting are covered in Appendix B.

The properties of the MLP are discussed in the following sections, and its performance compared with those of standard statistical classifiers.

Multi-Layer Perceptron

We have adopted the standard MLP network architecture (Rumelhart, 1986). Processing units are arranged in layers - an input layer, a number of hidden layers (usually one or two) and an output layer. There is full connectivity between adjacent layers and no connectivity between non-adjacent layers.

Each processing unit (i.e. excluding input nodes) performs a weighted summation of its inputs, one of which, the bias, can be considered as a weight from a dummy unit whose output is always one. A non-linear function is then applied to the total summation. More formally, let

y_i denote the output from neuron i,
x_j denote the input to neuron j,
W_{ij} denote the weight between nodes i and j,

The input to node j is given by

$$x_j = \sum_i y_i W_{ij}$$

and its output by the transfer, or response, function $F(x_j)$ which can be linear, but is more normally a sigmoid function

$$F(x_j) = \frac{1}{1 + e^{-x_j}}.$$

The MLP is trained by repeated presentation of a series of input patterns and their corresponding target classes. The mapping from input to target space is learned by minimising

$$E = \frac{1}{2} \sum_{p=1}^{n} (\underline{y}_p - \underline{d}_p)^2$$

where $\underline{y}$ and $\underline{d}$ represent the observed and target vectors respectively, and n is the number training vectors.

The learning rule is given by

$$\Delta W_{ij}(t) = -\epsilon \frac{\delta E}{\delta W_{ij}(t)} + \alpha \Delta W_{ij}(t-1)$$

where t denotes the iteration, ϵ is the learning rate i.e. step size, and α, the momentum, enforces the degree of similarity between consecutive weight changes. The momentum and learning rate values recommended by Rumelhart are 0.9, and 0.1 respectively, and these have been adopted in the experiments that follow.

Data Fusion Using an MLP

For a binary classification network, Dodd (1990) proved that, subject to certain conditions, squared error minimisation can cause outputs to approximate probabilities (see Appendix C). These conditions are met provided that the output is an unbiased estimate of the probability (say a 1 from c coding) and that the training data is a representative sample of the pattern space. The quality of this approximation may sometimes be degraded by inadequate architecture or convergence to local minima, both of these problems are reduced by the usual techniques of searching for suitable network architectures and repeated training from random starts. Two questions remain unanswered. Can an MLP fuse correlated opinions that are given in the form of measures of belief, and if so, are the results probabilities? We would expect the answer to both of these questions to be "yes". An MLP should be capable of *learning* both the correlations between expert opinions, and the distribution of their respective expressions of belief. Thus, provided that Dodd's constraints are satisfied, the outputs should be probabilities regardless. This leaves aside the issue as to whether or not probabilities or beliefs are a sufficient description of the results from previous classification systems. This will be addressed in future research (Appendix D).

The MLP's data fusion capability was tested by simulating a sensor system consisting of three independent channels. The image from each channel was represented by a different pair of Brodatz (1966) textures bounded in each case by a diagonal line running from the top left to the bottom right hand corner. Each image was processed in turn by a texture discrimination program resulting in three grey level encoded posterior probability maps. These images can be viewed as representing the evidence produced by three vision modules, each of which relates to a different channel.

The MLP was trained using 240 off-diagonal image blocks, with each training vector consisting of three input probabilities, one from each channel, together with a target class label (0 or 1). The resulting vectors were used to train the "general purpose MLP" program developed by Dodd (1988). Details of specific network architectures will be discussed in the following section.

The trained MLP was tested on the probability images shown in figures 1a,b and c. The types of texture used in their construction correspond with those used in the training set, though the samples are different. The most notable observations are that expert "a" is least reliable, and that the labelling convention adopted by expert "b" is not agreement with its counterparts. The MLP output is shown in figure 1d. Thresholding at the 0.5 probability level reveals 9 classification errors in the fused segmentation, while the input channels contain 65, 35, and 31 errors respectively.

The more general problem of combining correlated measures of belief was tested by training the MLP with linear combinations of the original probability maps, whose distributions have been subsequently scaled and offset by differing amounts. The test and the fused images are shown in figure 2. Comparing figures 1d and 2d shows that when fused, the original and correlated channels produce near identical results.

It could be argued that instead of examining single image blocks, greater coherence would be obtained by considering the surrounding blocks in say a 3x3 neighbourhood. However, problems arise in the vicinity of boundaries, where, for the fusion mechanism to be reliable, it should be trained with examples of each boundary orientation, for each possible position of the neighbourhood mask over the boundary. Here it is preferred to circumvent the problem by using single blocks, while incorporating contextual knowledge by including a separate post-processing stage (along the lines of that used by Chou and Brown (1987a,b & 1988)).

MLP Architectures

MLPs with various node configurations have been tested. The trained MLP was tested by classifying both the training and test sets. One would expect the classification error for the training set to decrease as the number of nodes increases i.e. the decision boundary becomes more complex. Applied to the test set, performance should increase until some optimum configuration is found, beyond which the decision boundary is modelling noise and consequently loses its ability to generalise. In each case the MLP was trained for 100000 iterations of the complete training set (240 samples), with the weights being updated after each iteration. This amount of training might be considered excessive but it almost ensures that the optimisation algorithm reaches a minimum, global or otherwise. Each MLP configuration was executed about 10 times from different random weight starts. Results are tabulated for both independent and correlated channels (Table 1).

Table 1. MLP architecture performance (ten random weight starts)

image data	**network config.**	**transfer function**	**training**	**testing**	
			lowest $\sum$ **error**^{2}	$\sum$ **error**^{2}	**classification error**
independent	3I-1O	linear	4.91	13.29	7.08%
	3I-1O	logistic	0.96	9.39	3.75%
	3I-2H-1O	logistic	<0.01	10.21	4.58%
correlated	3I-1O	linear	4.90	13.41	7.92%
	3I-1O	logistic	0.90	8.89	3.33%
	3I-2H-1O	logistic	0.80	10.48	5.00%

For the problem above, it appears that the best architecture has three input nodes, no hidden nodes (though one has been masked) and one logistical output node. This happens to be the simplest network that produces outputs guaranteed to look like probabilities (i.e. in the range 0 to 1). It is reasonable to suppose, however, that performance might be improved by adjusting other parameters including learning rate, momentum and the transfer function. Since the performance achieved is acceptable, and there is no guarantee that any fine tuning will be applicable to other problems, this architecture will be adopted in future experiments.

Comparison with Standard Classifiers

To determine whether or not the MLP is learning a mapping that is readily determinable by conventional methods, its performance is compared with those of standard classifiers. Of particular interest are the nearest class mean (NCM) and k nearest neighbour (KNN) classifiers.

The nearest class mean classifier assigns the unknown input vector, $\overline{x}$, the label corresponding to the closest class mean. The distance metric is often Euclidean, however, if sufficient training samples are available for computing a reliable covariance matrix for each class, then the Mahalanobis distance M is usually preferred. Its advantage lies in its ability to take into consideration the correlations between variables. The Mahalanobis distance between a single multivariate observation and the centre of a population is given by

$$D_i = (\overline{x} - \overline{\mu_i})^t \Sigma_i^{-1} (\overline{x} - \overline{\mu_i})$$

where, for class i, $\overline{\mu_i}$ is the mean feature vector, and $\sum_i^{-1}$ is the inverse covariance matrix.

Briefly, the k nearest neighbour classifier operates by constructing a hyper-sphere about $\overline{x}$ which contains k training samples. From Duda and Hart (1973), k should be odd and $\simeq \sqrt{n}$, where n is the total number of training samples. If, out of the k training samples closest to $\overline{x}$, k_i of them belong to class ω_i, then an estimate of posterior probability is given by $P(\omega_i|\mathbf{x}) = k_i/k$. The k nearest neighbour algorithm relies on the accurate computation of Euclidean distance between pairs of points in a high dimensional space. Associated with this are two problems, firstly that of the unequal variability of features, and secondly, that the features may not be orthogonal. The former is often alleviated using standardisation, however, this is only appropriate if the spread of values is due to normal random variation and not the presence of subclasses. Similarly, principal component analysis is often used to counter the effects of correlated features.

The performances of the MLP, NCM and KNN classifiers are compared. The KNN classifier is tested on raw, standardised and orthogonalised feature values.

Table 2. MLP, NCM and KNN performances

image data	**classification error and sum squared error rates**				
	NCM	**KNN**			**MLP**
	(M'obis)	**raw**	**stand.**	**orthog.**	(3I-1O,logistic)
independent	7.5% 18.0	5.0% 10.8	5.0% 9.9	5.4% 10.8	3.8% 9.4
correlated	7.5% 18.0	11.3% 19.9	10.0% 19.7	5.8% 11.2	3.3% 8.9

The advantage of the NCM classifier is its immunity to the effects of correlated features, however, it has the disadvantage of being parametric and of not producing classifications in the form of posterior probabilities. The KNN classifier is non-parametric and theoretically, when supplied with an infinite amount of training data, its performance is optimal. In practice, however, the KNN classifier relies on having an effective distance metric, and consequently performance drops when features exhibit widely differing variances or are highly correlated. Attempts to alleviate these problems by standardisation or principal component analysis can be effective in certain situations, but in general they cannot be relied upon. MLPs are subject neither to the restrictions of a parametric model, nor to the problems arising from the computation of distances in feature space. Although there are situations when the KNN classifier will out perform an MLP, it is the MLPs general purpose nature that makes it desirable.

Conclusions

A multi-layer perceptron has been applied to the data fusion problem. The desirable properties an MLP can be summarised as follows :

- it is robust to differences in expert reliability;
- it is sensitive to correlations between experts;
- the opinions supplied to the MLP do not have to be probabilistic;
- The outputs of a binary classification network can be interpreted, under certain conditions, probabilities.

Contextual information could be included in the fusion process by using local neighbourhoods. Alternatively, a probability map generated by an MLP could be combined with prior knowledge (modelled as a Gibbs distribution) by using a Bayes estimator. Thus spatial coherence between adjacent image blocks can be enforced, and any prior knowledge of the likely arrangements of image features can be utilised.

Acknowledgements

The authors gratefully acknowledge the use of Nigel Dodd's GMLP package. Thanks also to John Bridle, Mark Bedworth, Nigel Dodd, Paul Lloyd, Steve Lutrell and Simon Shannon for providing useful input.

Appendix A: Potential Problems Due to Inconsistent Prior Information.

This problem can be rigorously considered in a Bayesian/Likelihood framework. It is quantitatively valid to combine probabilities from likelihood estimates of a parameter, the simplest case for independent data being to multiply the probabilities

$$P(\overline{x}|\omega) \;=\; P(x_1|\omega)P(x_2|\omega)$$

Where ω is the classification hypothesis, with x_1 x_2 being two statistically independent sets of data. We can then combine with a prior, if we wish to give an a-posterior probability;

$$P(\omega|\overline{x}) \;=\; P(\overline{x}|\omega)P(\omega)$$

However, simple multiplication of a-posterior probabilities will lead to double counting of the prior;

$$Q \;=\; P(\omega|x_1) \times P(\omega|x_2) \;=\; P(\omega|\overline{x})P(\omega)$$

In fact, fusion of the results from algorithms which incorporate prior information requires us to first un-do the effects of the prior in order to produce the objective estimate of parameters based upon the evidence, (this requires knowledge of what the prior was). These unbiased estimates can then be combined. Data based upon inconsistent prior information cannot be used directly to produce a quantitatively valid ('honest') estimate of probability.

This simplest of examples therefore illustrates that algorithms which use a Bayesian approach to constrain solutions but do not make available the assumed prior for later stages of processing would be expected to lead to problems when attempting fusion of these results with those from other modules. If the assumed priors are also implicit within an algorithm it would be difficult to know if they are consistent with those assumed in other modules. These and other issues regarding quantitative use of Bayes theory are further discussed in Tina Memo 2002-005. Use of an MLP for data fusion can be shown to solve these problems, at least for cases where a representative data sample is available for training (Appendix C).

Appendix B: Boosting and its Problems.

Boosting is a recent suggestion for fusing opinions of multiple simple classifiers. There are many variations and extensions possible, so in order to keep this discussion general we will define boosting as any combination process which performs a linear weighting of decisions from classifiers where each is trained on sub-sets of data selected by one or more of the other simple classifiers. Although this technique may be thought to be a recent innovation, other authors have suggested equivalent methods previously (see Kittler et.al. BMVC 1997), with equivalent limitations. The basic idea can be traced back as far as the Perceptron (see below).

In order to make progress we must relate the boosting process to an established statistical method and identify the assumptions required for the technique to be valid. In this respect majority voting can be considered the same process as hypothesis testing. Weighted combination is then valid if we interpret the outputs from each simple classifier as a log-probability, where each may be correlated globally with the others. Clearly however, if the decision rules for one classifier have been determined by weighting data according to the performance of another classifier then the data is not global but local and so therefore the combination process is inappropriate.

example: Imagine, that we are building a classification system and start by building one simple classifier. In order to build a second simple classifier we then weight the selection of a subset of the training data set in favour of majority classification error. Clearly this second rule partitions this subset of data in the direction of the classification as intended, but what about the original (full) sample? Do the examples which have now been preferentially excluded also separate favourably on the basis of this feature? If the answer is no then any advantage in classification gained by adding the new feature for the data sub-set may be undone when applying a weighted decision to the full data set.

Avoiding this problem requires that the use of later simple classifiers should be conditional on the decisions made by earlier classifiers (ie: applied only to appropriate local regions of the data space), similar to the decision trees found in early papers (circa 1970) on expert systems, such as ID3. The problem with this particular approach is that it turns our classifiers into binary classifiers (see Appendix D).

The weighted voting combination process is therefore entirely dependent upon a chance distribution of data density and cannot be guaranteed a-priori. It may work for some datasets, but not for others. It is certainly not optimal use of the data. In fact this anti-correlation example is analogous to the X-OR problem (Minsky and Papert) which killed most research on the Perceptron. In other words, the class of problem which the Perceptron could not deal with are poorly handled by Boosting, and if this killed the Perceptron in 1968 ......?

The way to solve this problem requires the ability to learn the (potentially local) correlations between outputs from the simple classifiers. This is precisely what using an MLP for the task achieves. Moreover, this approach is not restricted to purely global linear correlations, as modelled by techniques such as PCA.

Appendix C: Squared Error Minimisation Causes Output to be Probabilities.

Consider the binary classification of an unknown vector $\overline{x}$. The output, $f(\overline{x})$, of an MLP having a single output node is targeted at 0 for class ω_0 and 1 for class ω_1. The sum-squared error, E, over the training set will then be

$$E = \int_{-\infty}^{\infty} P(\omega_0)p(\overline{x}|\omega_0)[f(\overline{x}) - 0]^2 + P(\omega_1)p(\overline{x}|\omega_1)[f(\overline{x}) - 1]^2 d\overline{x} \quad - \quad (1)$$

where $P(\omega_i)$ is the prior probability of class ω_i, and $p(\overline{x}|\omega_i)$ is the probability function for $\overline{x}$ given class ω_i. If the MLP is trained to minimise the sum-squared error (which is the usual training prescription), it will fashion f accordingly (barring local minima and assuming sufficient network architecture). Differentiating E with respect to the function f gives

$$\frac{\partial E}{\partial f} = 2p(\overline{x}|\omega_0)P(\omega_0)f(\overline{x}) + 2p(\overline{x}|\omega_1)P(\omega_1)[f(\overline{x}) - 1] \quad - \quad (2)$$

and equating this to zero

$$f(\overline{x}) = \frac{p(\overline{x}|\omega_1)P(\omega_1)}{p(\overline{x}|\omega_1)P(\omega_1) + p(\overline{x}|\omega_0)P(\omega_0)} \quad - \quad (3)$$

which is exactly the probability of the correct classification being ω_1 given that the input was $\overline{x}$. So, training the network to minimise the sum-squared error, results in a probabilistic classifier. Substituting the $f(\overline{x})$ corresponding to a trained network back into (1) gives

$$E = \int_{-\infty}^{\infty} \frac{P(\omega_0)p(\overline{x}|\omega_0)P(\omega_1)p(\overline{x}|\omega_1)}{P(\omega_0)p(\overline{x}|\omega_0) + P(\omega_1)p(\overline{x}|\omega_1)} d\overline{x} \quad - \quad (4)$$

the minimum error to be expected given the distributions of ω_0 and ω_1.

This property of MLP training was also independently identified by Richards and Lippman (1991). The same ideas have since been reinvented by Platt et al. (1999) as methods for correcting the outputs of support vector machines and Boosting (Niculescu-Mizil, 2005), in order to generate honest conditional probabilities of class and improve classification performance [4].

Appendix D: Theory of Probability Fusion.

We wish to understand the limits of fusing outputs from low level vision modules, and in particular what it would mean to perform this process in an optimal manner. Assuming that the low level modules deliver probabilities, for example $P(\omega|x_1)$ and $P(\omega|x_2)$, then the process of training an MLP to fuse these values can be considered as learning the mapping function F_ω for;

$$F_\omega(P(\omega|x_1), P(\omega|x_2)) \;=\; P(\omega|P(\omega|x_1), P(\omega|x_2))$$

Clearly, the optimal way to use the data x_1 and x_2 would be to compute

$$P(\omega|x_1, x_2)$$

we generally must abandon any hope of computing this quantity due to the volume of the pattern space covered by all possible values of x_i. This is normally the justification for taking the modular approach and attempting

[4] Some readers might be suprised to hear that it is neccesary to bolt the equivalent of a neural network onto the back of a Boosted or SVM classifier in order to get them to work properly.

subsequent fusion. We can say that these two approaches are equivalent only if the information content in the original data is still present in the probabilities; ie:

$$P(\omega|P(\omega|x_1), P(\omega|x_2)) \;=\; P(\omega|x_1, x_2)$$

Or put another way, that the original data can be theoretically regenerated (but for any unwanted invariances) from the probabilities. This concept can be referred to as the vector $(P(\omega|x_1), P(\omega|x_2), .., P(\omega|x_i))$ being a "complete" representation of the data vector $(x_1, x_2, ..., x_i)$. We can see that this is unlikely to be possible with a system which recognises only a small set of data types ω, as the encoded representation will simply not have enough degrees of freedom to capture the dimensionality of the original data. It is however, very likely to happen if we have a sufficiently rich set of competing hypothesis categories with which to represent the data in the region of ω.

This analysis is also valid for replacement of the $P(\omega|x_i)$ with any monotonic function, lending theoretical validity to a wide variety of approaches. However, its also illustrates the importance of working with continuous valued outputs from our low level modules, binary output quantities should clearly be avoided. This appendix has been largely derived from the work presented by Ian Poole at BMVC 1990, who shows that the mapping process satisfies the above requirements if the data to be fused has the property of Class Conditional Independence (CCI).

References

Berenstein, C., Kanal, L.N. and Lavine, D. (1985) "Consensus and evidence," in *Pattern recognition in practice II*, Edited by Gelsema, E.S. and Kanal, L.N., Elsevier Science Publishing Company, inc.

Booth, D.M. and Mayhew, J.E.W. (1989b) "The adaptive self parameterising texture region tracker," *Proc. 5th. Alvey Vision Conference,* University of Reading, 25th - 28th September.

Booth, D.M. and Mayhew, J.E.W. (1990a) "Multi-spectral texture analysis," RIPREP 1000/73/90, RIPR, DRA Electronics Division, St.Andrews Road, Malvern, Worcs.

Booth, D.M. and Mayhew, J.E.W. (1990b) "Boundary tracking on the basis of stereo disparity" RIPREP 1000/76/90, RIPR, DRA Electronics Division, St.Andrews Road, Malvern, Worcs.

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

Chou, P.B. and Brown, C.M. (1987a) "Probabilistic Information Fusion for multimodal image segmentation," *Proc. International Joint Conf. in Artificial Intelligence*, Milan, Italy, August.

Chou, P.B. and Brown, C.M. (1987b) in *Proc. Image Understanding Workshop*, Los Angeles, February.

Chou, P.B. and Brown, C.M. (1988) in *Proc. Image Understanding Workshop*, Cambridge, April.

Dodd, N. (1988) "General Purpose Multilayer Perceptron Program," RIPR report 1000/40/88, Royal Signals and Radar Establishment, St. Andrews Road, Malvern, Worcs.

Dodd, N. (1990) "Intensive care ward monitoring using 'default' training," *Int. Conf. Neural Networks*, Paris.

Duda, R.O and Hart, P.E. (1973) *Pattern classification and scene analysis*, Wiley and Sons.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) "Learning internal representations by error propagation," In *Parallel Distributed Processing Systems, Denver, 1988,* pp.305-313.