# Neural Networks for the Texture Classification of Segmented Regions of Forward Looking Infrared Images

John. F. Haddon*, James. F. Boyce†
Simon Protheroe†, Simon Hesketh†
* Defence Research Agency, Farnborough, Hampshire, England, GU14 6TD
† Wheatstone Laboratory, King's College, Strand, London, WC2R 2LS


**Abstract**

Texture can be interpreted as a measure of the 'edginess' about a pixel and can thus be described by edge co-occurrence matrices. The matrix can be decomposed using 2-dimensional orthogonal Hermite functions, the coefficients of which provide a low order feature vector which is characteristic of the texture. The Hermite coefficients for 240 hand-segmented regions of grass, trees, sky and river from 60 forward looking infrared (FLIR) images have been used to train and validate 2 neural networks, which have subsequently been used to label FLIR images segmented using co-occurrence techniques [1].

## 1 Introduction

Almost any image processing system will initially analyse an image to determine its constituent regions and/or the major boundaries between these regions; and will then analyse the processed image to determine basic properties about the segmented regions, usually based on the boundary or internal areal features, such as measures of the region's grey level or texture. Co-occurrence matrices [2] are a recognised way of describing texture and statistical descriptors of the matrices are frequently used as the basis for textural region classification. Unfortunately, it is not trivial to select the appropriate descriptors and, furthermore, most descriptors only perform well on high quality images in low noise situations. Most techniques also make the implicit assumption that the regions being analysed are composed of a single texture. Past work by the authors has resulted in techniques [1] based on edge co-occurrence matrices (described in section 2) and the work described here has built on these techniques for region classification. The goal is the classification of regions of a FLIR image into terrain classes of grass, trees, sky or river.

The underlying structure of most edge co-occurrence matrices is Gaussian due to the predominantly Gaussian nature of the noise in the images being considered. If this underlying structure is removed, then the remaining structure is due primarily to the texture in the region: it is this structure that is exploited in the development of a texture classifier and makes the techniques developed robust compared to those currently available, such as that due to Haralick [2].

Any function can be decomposed to an arbitrary accuracy using a set of orthogonal functions; the error in the decomposition will decrease, or at worst remain the same, as the number of orthogonal functions used in the decomposition is

BMVC 1993 doi:10.5244/C.7.20

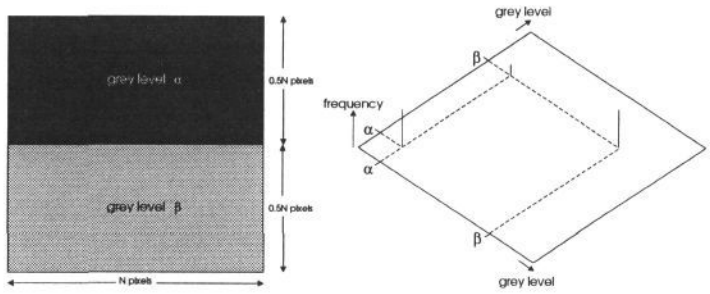


Figure 1: a) An image of 2 grey levels, b) The co-occurrence matrix

increased. In this paper, edge co-occurrence matrices will be decomposed using 2-dimensional discrete Hermite functions. These have been chosen because their basis function is Gaussian and therefore the zeroeth term in the decomposition will describe the noise while higher order terms describe the structure of the matrix and hence the texture of the region. Section 3 defines 1-dimensional discrete Hermite functions and shows how they can be used for describing a 1-dimensional function. Section 4 extends these ideas to 2-dimensions while section 5 uses them for decomposing edge co-occurrence matrices (examples are given for two regions of different terrain types: grass and trees). There are clear visual differences in the coefficients used in the description of these two visually similar classes. The coefficients of the decomposition provide a feature vector which is descriptive of the texture of the region.

The design of a classifier is a basic problem of pattern recognition for which there is no known general solution. The problem is one to which neural networks have been applied successfully in many instances. Section 6 describes the applications of two standard algorithms to the texture classification problem: a multilayer perceptron network and a cascade correlation network respectively. The input vector is formed from the decomposition of the edge co-occurrence matrices. Although a complete description of the matrices requires an infinite set of functions, it is clearly not practical to use a very large input vector: not only would the use of such a large vector be uneconomical but the design of a classifier would be difficult and the resulting classifier would be fragile unless a very large set of training data were available. The Kolmogorov-Smirnov statistic is used to determine which are the most important features. The top $N$ features for each pair of classes are selected, these are calculated for all classes and input to the neural networks. The networks are trained using regions of grass, trees, sky and river from 60 FLIR images: An independent validation set is used to prevent overtraining. The architecture of the networks, the number of hidden layers and the corresponding number of nodes, is varied to optimise the generalisation properties of the networks.

## 2 Co-occurrence Matrices

A co-occurrence matrix $S$ [3] is basically a 2-dimensional histogram in which each element $i, j$ is the frequency (sometimes normalized) with which two events $i$ and $j$ co-occur with a specific relationship to each other. As a simple example, consider the element $i, j$ in the grey level co-occurrence matrix, $S_{\Delta}(i, j)$, the value of this element is the frequency with which grey level $i$ occurs at a displacement $\Delta$ relative to grey level j. For nearest neighbour matrices, $\Delta$ is one of $(0, 1), (1, 0), (-1, 0)$ or $(0, -1)$.

Figure 1 a is an image consisting of two regions of uniform grey levels, $\alpha$ and $\beta$, separated by a horizontal boundary. Within the lower region, two vertically adjacent picture elements or 'pixels' will both have grey level $\alpha$. The value of the $(\alpha, \alpha)$ element of the co-occurrence matrix (figure 1b) will be the number

of times that two vertically adjacent pixels both have the grey level $\alpha$, *ie* the width of the region in pixels, $n$, times its height in pixels less 1, $(\frac{n}{2}-1)$. Hence, $S(\alpha,\alpha) = n(\frac{n}{2}-1)$ Similarly, the upper region yields a contribution $n(\frac{n}{2}-1)$ at $(\beta,\beta)$. The value of the matrix at position $(\alpha,\beta)$ will be the number of times a pixel pair straddles the boundary between the two regions, *ie* $S(\alpha,\beta) = n$, the length of the boundary, Although there is a boundary between regions with grey levels $\alpha$ and $\beta$ respectively, there is no corresponding boundary between regions with grey level $\beta$ and $\alpha$, hence there is no contribution to the matrix at position $(\beta,\alpha)$.

The locations of the region distributions, at $(\alpha,\alpha)$ and $(\beta,\beta)$, in figure 1b imply the location of the boundary distributions at $(\alpha,\beta)$ and $(\beta,\alpha)$. The distance from a boundary distribution to the leading diagonal of the matrix is proportional to the edge strength $E$ between the two regions *ie* the convolution of the two pixels with the first order difference operator $[1\ -1]$. In this example, the edge strength is

$$E = \frac{\beta-\alpha}{\sqrt{2}} \tag{1}$$

The effect of Gaussian noise in the image will be to spread the distributions in the co-occurrence matrix into two-dimensional Gaussians. The effect of texture in the image will be to spread the distributions along the leading diagonal of the matrix.

The close relationship between the co-occurrence matrix and an edge operator is emphasised by terming this type of matrix an *edge* co-occurrence matrix. More general edge operators (such as a Canny [4]) can easily be used in the formation of the matrix. Consider an $L \times L$ image of $N \times N$ pixels where $N$ is even and each pixel is of dimension $\Delta x \Delta y$, and indexed by

$$\mathbf{x} = (x,y) = (m\Delta x, n\Delta y); \quad -\frac{N}{2}+1 \leq m,n \leq \frac{N}{2} \tag{2}$$

Let the intensity $i(\mathbf{x})$ of pixel $\mathbf{x}$ lie in the range $1 \leq i(\mathbf{x}) \leq I$. An edge operator $E_{\boldsymbol{\Delta}}$, such as Spacek's [5] realization of the Canny, forms an edge strength, $e_{\boldsymbol{\Delta}}(\mathbf{x})$, from an intensity image $i(\mathbf{x})$ by convolution,

$$e_{\boldsymbol{\Delta}}(\mathbf{x}) = \left(E_{\boldsymbol{\Delta}} * i\right)(\mathbf{x}) = \sum_{\alpha=-n}^{n} E_{\boldsymbol{\Delta}}(\alpha) i(\mathbf{x}+\alpha\boldsymbol{\Delta}) \tag{3}$$

where $\boldsymbol{\Delta}$ defines the direction of the operator. The edge image is then obtained by combining the edge strengths corresponding to different directions. The operator has $(2n+1)$ elements, while, since it is based on a first order derivative, the coefficients are anti-symmetric. It thus divides naturally into left and right components and is a natural basis for the formation of the edge co-occurrence matrix,

$$S^E_{\boldsymbol{\Delta}}(i,j) = \sum_{\mathbf{x}\in P} \delta\left(i; \sum_{\alpha=1}^{n} E_{\boldsymbol{\Delta}}(\alpha) i(\mathbf{x}-\alpha\boldsymbol{\Delta})\right) \delta\left(j; \sum_{\alpha=1}^{n} E_{\boldsymbol{\Delta}}(\alpha) i(\mathbf{x}+\alpha\boldsymbol{\Delta})\right) \tag{4}$$

Texture can often be thought of as a descriptor of *edginess* in an image, possibly a group of edges occur at some regular displacement, such as in an image of bricks and mortar. The orientation and periodicity of the texture may be important in its characterisation and can be determined by Fourier techniques [6].

Figure 2a shows a typical 2-dimensional Spacek operator while figure 2b shows the same operator with a displacement from the origin and a rotation. This operator would be sensitive to edges similar to those of the operator shown in figure 2a but with some 'spatial' displacement between them, such as may occur in an image of bricks and the mortar between them. This type of operator can

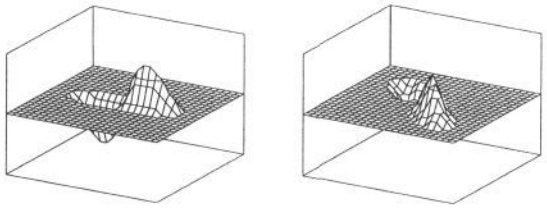

Figure 2: a) A Spacek edge operator, b) A Spacek edge operator with a 'zero' central part

Figure 3: The first 6 one-dimensional Hermite functions (not to the same scale)

be used in the formation of a co-occurrence matrix in much the same way as any other operator.

Each component distribution of an edge co-occurrence matrix is a two- dimensional function composed of a Gaussian (due to Gaussian noise in the image) with higher level structures caused by the region's texture. Any function can be decomposed using a set of orthogonal functions, the coefficients of which can be used to reconstruct the original function. The error in the reconstruction will always decrease (or at worst remain the same) as the set of orthogonal functions is increased. The underlying form of an edge co-occurrence matrix is Gaussian and is therefore an appropriate basis function for the definition of a set of orthogonal functions: such as Hermite functions. The discrete nature of the problem means that the orthogonal functions should be defined in a discrete manner.

# 3 Discrete Orthogonal Hermite Functions

Consider the discrete function $f(x) \equiv f(n\Delta x)$, $-N \leq n \leq N$. Discrete Hermite functions $\Phi_l, l = 0, 1, ...,$ can be defined to satisfy $(\Phi_l, \Phi_m) = \delta_{lm}$ where $\delta_{lm} = 1$ if $l = m$ and 0 otherwise with respect to the inner product defined as

$$(\Phi_l, \Phi_m) = \sum_{n=-N}^{N} \Phi_l(n\Delta x)\Phi_m(n\Delta x)\exp(-n^2\Delta x^2) \tag{5}$$

Let the zeroeth order Hermite, $\Phi_0 = a$ so that

$$a^2 \sum_{n=-N}^{N} \exp(-n^2\Delta x^2) = 1 \tag{6}$$

Higher order Hermites will take the form

$$\Phi_l = \sum_{i=0}^{l} \alpha_i (n\Delta x)^i \tag{7}$$

where $\alpha_i, i = 0, .., l$ can be determined by Schmidt orthogonalization.

Figure 3 shows the first 6 one-dimensional discrete Hermite functions (not to the same scale). Any function can be decomposed using various proportions of these (and higher order) functions in a way similar to the Fourier expansion of a signal of finite duration.

Figure 4a shows a test function of a Gaussian distribution centred on the origin and a secondary Gaussian distribution to one side. The test function can be decomposed by adding together various proportions of the Hermite functions

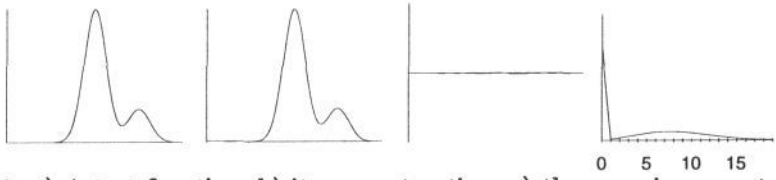


Figure 4: a) A test function, b) its reconstruction, c) the error in reconstruction, d) the Hermite coefficients

(shown in figure 3). The coefficients used in the decomposition are shown in figure 4d while the reconstruction is shown in b. There is a high coefficient corresponding to the zeroeth order Hermite function which describes the underlying Gaussian at the centre. The remaining coefficients describe the secondary peak. Although an infinite set is required for a perfect decomposition, a reduced set can be used. In this decomposition, the first 20 Hermite functions were used in this test and the error is negligible (figure 4c).

In a similar way, two-dimensional functions (such as an edge co-occurrence matrix) can be decomposed using two-dimensional discrete Hermite functions.

# 4 Decomposing Co-occurrence Matrices Using Discrete Hermite Functions

A two-dimensional function $f(n\Delta x, m\Delta y)$, centered at $(x_0, y_0)$ with support $-N \leq l, m \leq N$, can be decomposed into $pq$ discrete orthogonal Hermite functions as

$$f(n\Delta x, m\Delta y) = \sum_{l=0}^{p}\sum_{k=0}^{q} f_{kl}\Phi_k(n\Delta x - x_0)\Phi_l(m\Delta y - y_0) \exp\left(-\left(\frac{n\Delta x - x_0}{2\sigma_x}\right)^2 - \left(\frac{m\Delta y - y_0}{2\sigma_y}\right)^2\right) \quad (8)$$

where

$$f_{kl} = \sum_m \sum_n f(n\Delta x, m\Delta y)\ \Phi_k(n\Delta x - x_0)\Phi_l(m\Delta y - y_0) \exp\left(-\left(\frac{n\Delta x - x_0}{2\sigma_x}\right)^2 - \left(\frac{m\Delta y - y_0}{2\sigma_y}\right)^2\right) \quad (9)$$

with an error $\eta_{p,q}$ in the expansion of

$$\eta_{p,q} = \sum_m \sum_n \left[f(n\Delta x, m\Delta y) - \sum_{k=0}^{p}\sum_{l=0}^{q} f_{kl}\Phi_k(n\Delta x - x_0)\Phi_l(m\Delta y - y_0) \exp\left(-\left(\frac{n\Delta x - x_0}{2\sigma_x}\right)^2 - \left(\frac{m\Delta y - y_0}{2\sigma_y}\right)^2\right)\right]^2 \quad (10)$$

Since

$$\eta_{p,q} = \sum_{k=p'}^{\infty}\sum_{l=q'}^{\infty} |f_{k,l}|^2, \qquad (p' \geq p, q' \geq q) \quad (11)$$

then this error will remain the same, or decrease, if additional terms are used in the expansion, *ie*

$$\eta_{p',q'} \leq \eta_{p,q}, \ \forall (p' \geq p, q' \geq q) \quad (12)$$

The coefficients $f_{kl}$ are a feature vector from which the original function can be reconstructed to accuracy $\eta_{p,q}$. This feature vector is characteristic of the function being decomposed. The above equations assume axes parallel to a square grid. Distributions in a co-occurrence matrix have axes along and perpendicular to the leading diagonal of a matrix. In decomposing a co-occurrence matrix, equations 8 to 10 are implemented with a 45° rotation of the axes.

Figure 5 shows the first few two-dimensional Hermite functions (not to the same scale) formed for a 45° rotation.

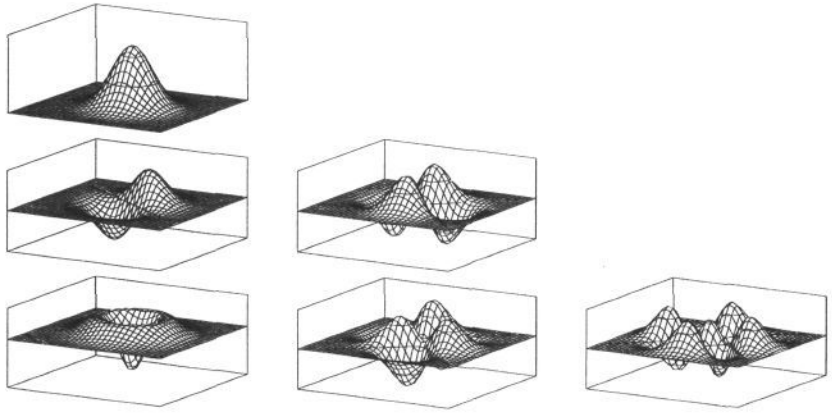

Figure 5: The first few two-dimensional Hermite functions (not to the same scale)

Figure 6: An infrared image of a bridge over a river with trees and gr' s

# 5 Texture Description using Discrete Hermite Functions

It was stated earlier that the underlying structure of an edge co-occurrence matrix is due to Gaussian noise in the image while the remaining structure of the matrix describes the texture of the region. When the matrix is decomposed using discrete two-dimensional Hermite functions, the zeroeth term is characteristic of the Gaussian noise while the higher order terms are characteristic of the texture.

Figure 6a shows a FLIR image of a bridge over a river taken from a low flying aircraft. A region of trees has been extracted from this image and the edge co-occurrence matrix (figure 7a) formed using an appropriate operator (based on analyses described in reference [6]). This matrix has been decomposed using the technique described above. Figure 7b shows an isometric plot of the reconstruction while c shows the error in reconstruction. Figure 8a is an isometric logarithmic plot of the coefficients of decomposition. The same processes have been applied to a region of trees and figure 8b is the corresponding plot of the Hermite coefficients

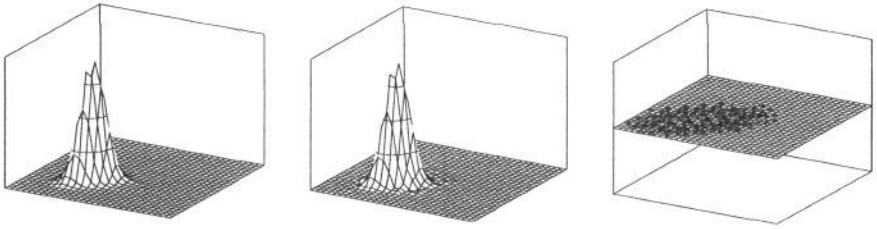

Figure 7: a) An edge co-occurrence matrix of grass, b) the reconstruction, c) the error in reconstruction

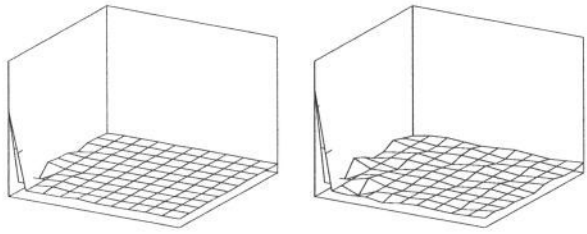

Figure 8: Logarithmic plots of the Hermite coefficients used in the reconstruction of the co-occurrence matrix of a)grass, b)trees

used in the decomposition of the matrix. There are very clear differences between the coefficients for trees and grass.

The Hermite coefficients are a feature vector which is characteristic of the texture of the region. It is likely that only a small subset of these coefficients contain sufficient information for separating the different textures. These coefficients can be determined using the Kolmogorov-Smirnov [7] (KS) statistic. Let the feature vector be $f_c(k), k = 1..K$ where $c$ is the class (grass, trees, sky or river) and $K$ is the total number of features (121 in the above examples). The KS statistic $E_{ij}(k)$ for any pair of classes $i$ and $j$ can be defined as

$$E_{ij}(k) = \sqrt{\frac{(\mu_i - \mu_j)^2}{\sigma_i \sigma_j}} \tag{13}$$

where $\mu_i$ and $\sigma_i$ are the within class mean and standard deviations. The $N$ coefficients with the largest KS statistic contain the most information and are selected for subsequent training and validation using neural networks.

# 6 Texture Classification using Neural Networks

Texture assessment data was obtained by hand-classifying a number of typical regions of various sizes from a data base of 300 images taken during a 12 second period as a low flying aircraft approached a bridge over a river (see figure 6). There were 240 regions in total (99 grass, 55 trees, 55 sky and 31 river). The data was divided into training and validation sets which were disjoint, so that validation was independent of training. The input vector is potentially very large (121 in the current experiments). Accordingly subsets of the feature space were determined using the KS statistic of equation 13.

## 6.1 Multilayer Perceptron Neural Nets

A multilayer perceptron partitions the feature space into a collection of disjoint regions by a set of hyperplanes. Each region corresponds to a distinct pattern class, though the same class may be represented by a number of disparate regions. The universal approximation theorem [8] guarantees that such a partition exists, providing that the pattern classes are indeed separate in the feature space. It does not construct the partition or indicate how many hyperplanes are required.

The simplest form of multilayer perceptron neural net consists of three layers (Figure 9a) termed the input, hidden and output layers respectively. Each layer comprises a collection of nodes: only interlayer connections are permitted between nodes. Information enters the net via the input layer which has a node and an input connection for each feature vector component. The number of nodes in the hidden layer determines the number of decision hyperplanes. The choice of the number of hidden nodes is a compromise between having sufficient nodes to adequately separate the training data but not so many that the classifier defined

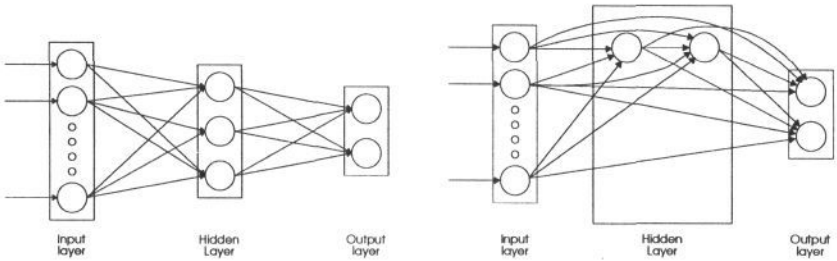


Figure 9: a) A multilayer perceptron, b) a cascade correlation neural network

by the net does not generalise to the test data. Information leaves the net via the output layer of nodes, one node and output for each pattern class.

A node having inputs $\{x_i, i = 1, \ldots, n\}$ has output

$$y = g\left(\sum_{i=1}^{n} a_i x_x + a_0\right) \tag{14}$$

where $g()$ is a sigmoidal function, monotonically increasing from 0 to 1. The set of weights of the node $\{a_i, i = 1, \ldots, n\}$ determines its response. The collection of the sets of weights of all of the nodes of the net determines its collective response and hence its memory. The weights are determined from the training data by the method of back error propagation [9]

## 6.2 Cascade Correlation Neural Nets

Like a multilayer perceptron, a cascade correlation neural network [10] has input, hidden and output layers of nodes. However the connectivity between them, as exemplified by Figure 9 is quite different from that of the former, though the net is still feed-forward.

Cascade-correlation was developed by Fahlman and Lebiere to overcome some of the limitations of learning algorithms designed for multi-layered networks, such as the back-propagation algorithm. The main difference with cascade-correlation is that the topology of the network is not fixed: it starts with a minimal net and trains automatically, adding new hidden units one by one, as they are needed. The new units are selected from a pool of candidate units trained in parallel; units with different output activation functions can be used. Each new unit, which forms a single-node hidden layer, receives a connection from each of the network's inputs and also from all pre-existing hidden units. Once a hidden unit has been created its input weights are frozen and only the output connections are trained. In this way, powerful high-order feature detectors are created. It is not unusual for very deep networks to be created, with a high fan-in to the hidden units. Figure 9 compares the architectures of multi-layer perceptrons and cascade-correlation nets. The multi-layer perceptron shown has only one hidden layer, with 3 nodes. The cascade-correlation net has created two single-node hidden layers.

## 6.3 Application of the Neural Networks

Subsets of the Hermite feature space were determined using the KS statistic. Selecting the best single, and best 5 features for separating each pair of classes resulted in 4 and 16 features being submitted to the neural network for training and validation. The data set contained 240 examples of which 99 were derived from grass, 55 from trees, 55 from sky and 31 from rivers. A validation set was formed by selecting every third data example, the remainder forming the training set. Three independent validation and training sets were so obtained. Figure 10

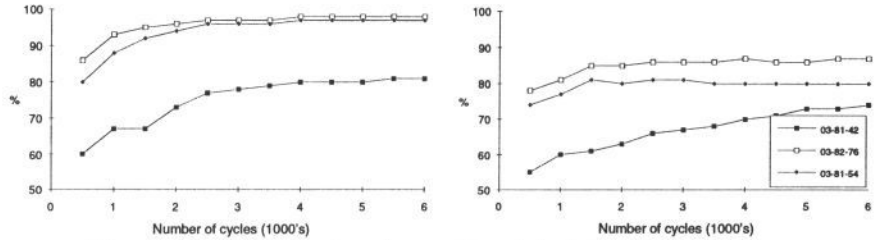


Figure 10: Neural network results: a) training, b) validation

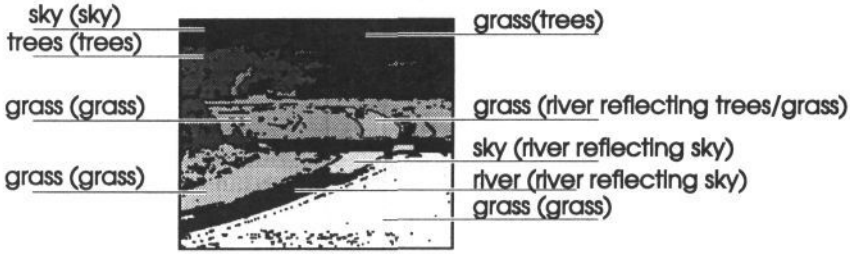


Figure 11: A labelled segmentation of figure 6 using the 16-10-4 multilayer perceptron (labels in brackets are the 'correct' ones)

show the training and validation results (means) obtained from the three independent trials for both the multilayer perceptron and the cascade correlation neural network.

The optimum performance of the multilayer perceptron was obtained from the 16-10-4 net (16 input nodes, 10 hidden nodes and 4 output nodes): 97% (training) and 85% (validation) with weights corresponding to 2500 training cycles. This configuration converged quickly to a stable state: in some of the other networks, the training performance continued to improve while the validation deteriorated as the number of training cycles was increased, indicating that the net was becoming overtrained, *ie* becoming attuned to the specific characteristics of the training set rather than the general nature of the pattern class of which it was a sample set.

The cascade correlation network was applied to the same data and achieved 100% on training and 85% on validation.

### 6.4 Example Application to Segmented FLIR images

Figure 6 was segmented using the techniques of reference [1] and the Hermite coefficients of the major regions determined and classified using the best multilayer perceptron (16-10-4), correctly classifying 6 out of 9 regions (figure 11). The neural networks were not trained on the river which reflected the hillsides and so errors have occurred here. Trees and grass are sometimes confused as has occurred in this example. Boundaries have been labelled black and some of the regions are too small to effectively determine the Hermite coefficients and hence can not be classified: these will be classified later using relaxation labelling techniques. These initial results are considered to be very encouraging.

## 7 Conclusions

This paper has presented techniques for decomposing edge co-occurrence matrices using 2-dimensional discrete Hermite functions. The zeroeth order Hermite describes the underlying Gaussian in the matrix and hence the noise in the image while the high orders describe the structure of the matrix and hence the texture of the region. There are significant visual difference in the coefficients of the Hermite

functions produced by the decomposition of matrices formed for regions of grass and trees. A subset of these coefficients, those containing the most information, was submitted to neural network classifiers. The difference in the Hermite coefficients for the different terrain classes is reflected in the success of the neural networks examined. In particular, the performance of the multilayer perceptron, achieving a high performance on training ($\approx$ 97%) while retaining reasonable performance ($\approx$ 85%) is considered to be very promising. The key difference between the techniques presented here and those of other researchers is that these techniques are robust under noise, this has been achieved because the zeroeth order Hermite describes the Gaussian noise while higher orders, which are used in the classification, describe the structure of the matrix and hence the texture of the region.

The best neural network has been applied to Hermite functions describing the co-occurrence matrices of segmented regions of a FLIR sequence. The results appear promising but further work needs to be done to ensure consistency in both spatial and temporal labelling, possibly using discrete relaxation labelling techniques. It may also be advantageous to use additional features in the neural network classifiers.

# References


[1] J F Haddon, J F Boyce, "Image Segmentation by Unifying Region and Boundary Information," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 12, No 10, pp929-948, 1990

[2] R M Haralick, K Shanmugan, Its'Hak Dinstein, "Texture Features for Image Classification", IEEE Trans. Sys. Man Cyber., SMC-3, pp610-621, 1973.

[3] R M Haralick, L G Shapiro, "Survey: Image Segmentation Techniques", Computer Vision, Graphics and Image Processing, Vol 29, pp100-132, 1985

[4] J F Canny, " A Computational Approach to Edge Detection", IEEE Patt. Anal. Machine Intell. **PAMI 8**, pp679-698, 1986

[5] L A Spacek, "Edge Detection and Motion Detection", Image and Vision Computing, **4**, pp43-56, 1986

[6] J F Haddon, J F Boyce, "Texture Segmentation and Region Classification by Orthogonal Decomposition of Cooccurrence Matrices", Proceedings of $11^{th}$ IAPR International Conference on Pattern Recognition, The Hague, The Netherlands, August $30^{th}$ to September $4^{th}$ 1992

[7] K. Fukunaga, "Statistical Pattern Recognition", 2nd Edn, Academic Press Inc., Boston, (1990)

[8] R Hecht-Nielson, "Kolmogorov's Mapping Neural Network Existence Theorem", Proc. Int. Conf. Neural Networks, Vol 3, pp11-13, IEE Press, New York

[9] D R Rumelhart, G E Hinton, R J Williams, "Learning Internal Representation by Error Propagation", Parallel Distributed Processing, eds. D E Rumelhart, J L McClelland, Vol 1, pp318-362, MIT Press 1986.

[10] S E Fahlman, C Lebiere, "The Cascade-Correlation Learning Architecture", Advances in Neural Information Processing Systems, ed. D S Touretzky, pp524-532, Vol 2, Morgan-Kaufmann, San-Meteo, USA 1990