The Stochastic Search Network

J.M.Bishop
Neural Network Research Group, Department of Cybernetics,
University of Reading, Berkshire, UK.

P.Torr
Robot Vision Group, Engineering Department,
Oxford University, Oxford, UK.

SECTION ONE : STOCHASTIC SEARCH

1.1 INTRODUCTION

A fundamental difficulty when using neural networks applied to pattern recognition, is the problem of stimulus equivalence - the invariance of symbolic information independent of transformation within a search space. For example, symbolically, the letter A remains an A irrespective of its position, size or orientation within an image field. Adult humans can generally recognise patterns accurately, despite such transformations and distortions. Classically it has been hypothesised that this ability is due to a normalisation process that occurs before the classification process begins.

One criticism of the network paradigm has been that there is no clear mechanism by which such normalisation can occur, since at first sight such unstructured systems do not provide any mechanism for attention - the focussing of analysis onto a subset of a larger pattern. This chapter will describe two network solutions to this normalisation process: Hinton Mapping and Stochastic Networks, give an introduction to the analysis of Stochastic Networks and illustrate their use within the British Telecom CONNEX project.

The problem of stimulus equivalence is a specific example of a best fit constraint satisfaction search, where the target pattern has been mapped into the search space, using translational, rotational or scale transformations. The goal of a system that can solve such problems is to ascertain the inverse mapping from search space to pattern. For example, in the situation where a letter has been rotated in a visual field, the inverse mapping would be the angle of rotation $\{\theta\}$, if it had been translated then the inverse mapping would be its cartesian coordinates $\{X,Y\}$, etc. One solution to this problem is a technique described by Hinton[1], summarised below.

1.2 HINTON MAPPING

A Hinton Mapping network consists of two sets of feature detectors, canonical and retinocentric (see figure one). Mutually excitory connections link the canonical

detectors with higher order detectors such as letter detectors, in the manner of the interactive activation and competition (iac) model[2].

Hinton describes a method for mapping retinocentric feature patterns into canonical feature patterns. Although the general application to three dimensional visual problems involves normalising six degrees of freedom, a simplified version which corrects for a one dimensional rotational transformation is described here.

Central to Hinton Mapping are fixed, *static*, mapping cells which implement potential mappings into the search space. There is one such cell *for each possible mapping into the search space*. Interactive activation and competition between cells ensures that the mapping which yields the best fit of the model into the search space will receive the most activation. After a few iterations, unless the model is symmetric, this mapping will become dominant.

Considering the situation in which a letter has been rotated clockwise by ninety degrees, a mapping system is required that will transmit activation from the activated retinocentric feature detectors to the relevant canonical detector, offset by a counter clockwise rotation of ninety degrees.

In figure one, three different mapping cells are shown, producing zero, plus ninety and minus ninety degree rotational shifts. Activity on mapping cell [x] produces one of two multiplicative inputs to a canonical cell, the other being from the corresponding retinocentric cell offset by [x] degrees. Thus for a canonical cell to receive input activation, a retinocentric cell must be active together with a mapping cell corresponding to the rotation of the retinocentric feature relative to the canonical feature. If only one of these cells is active, either a mapping cell or a retinocentric cell, then no activation will pass to the canonical cell. Thus the mapping cell effectively selects mappings from input space to normalised space.

If the correct rotational mapping is known in advance then it is possible to activate the corresponding mapping cell enabling the network to map from retinal to normalised coordinate space. Object recognition thus becomes:

i) Select a mapping, perhaps on the basis of prior knowledge.

ii) Activate corresponding mapping cell.

iii) Analyse degree of match between retinocentric stimuli and higher order features.

This process is repeated until the best match is found. The above technique is simply a sequential implementation of the standard normalisation and compare procedure and does not offer any improvement in performance over traditional methods.

The advantage of the above technique is that by using Hinton mapping cells it is possible to perform the search for the correct mapping in parallel. This is accomplished by using a further set of multiplicative connections to choose the correct mapping for a pattern given the retinal input and correct normalised pattern of activation.

Simultaneous activation of a central feature and a retinal feature is evidence that the mapping is correct. Thus, by allowing central and retinal cells that correspond under a specific mapping to project back to a multiplicative connection on the mapping cell, the activation of the mapping cell can be reinforced. Spurious conjunctions will occur, but correct mapping cells will in general receive more activation than the incorrect ones (unless there is ambiguity due to symmetry). By allowing mapping cells to compete, so that the ones receiving the most activation win, the network will eventually converge onto the correct mapping.

Initially when the retinal input is presented, all the mapping cells are partially active. Each retinal feature then stimulates partial activation to the canonical features corresponding to it over all mappings. The correct mapping allows the correct canonical cell to be partially active. Interactive activation between the canonical cells and the higher order cells will then reinforce the correct mapping relative to the background noise. This process alone can be sufficient for correct identification. However, the multiplicative connections on the canonical cells between the retinocentric cells and the mapping cells force further suppression of the noise. This process swiftly converges onto the correct interpretation of the pattern, as well as the correct canonical representation and correct mapping, purely from the retinal input.

However, examining the requirements of Hinton's scheme, we see that if the pattern is defined by a set of [M] features, and the search space consists of a set of [Z] features, then [Z] *static* mapping cells are required in the network. The number of mapping cells, [Z], will increase polynomially with the number of degrees of freedom of the the search. Thus in many situations, where the search space is very large, Hinton mapping is not a practical solution.

Stochastic Searching[3], a technique for solving constraint satisfaction problems, evolved from a study of Hinton Mapping, but only requires [M] mapping cells.

1.3 THE STOCHASTIC SEARCH

Stochastic Searching is a search technique which uses a random diffusion process to find the best fit of a given data model within a specified search space. The data model and the search space are defined in terms of Atomic Data Units (ADUs) between which comparisons can be made. For example, when searching for the best fit of a string of numbers within a larger string, the ADUs may be the digits that comprise the strings.

Unlike Hinton Mapping, where each mapping cell defines a fixed mapping, Stochastic Searching uses a network of variable (*dynamic*) mapping cells to test possible mappings of the model in the search space. Each cell defines one potential mapping of the model in the search space. There is one such cell for each ADU of the model. A random diffusion process ensures that mappings which appear successful are spread to other cells.

Stochastic searching is inherently parallel in nature; empirical evidence[4] suggests that the algorithm is probabilistically bounded by $O(Log(T_r))$, where T_r, the Time Ratio, is the ratio of the size of the model to the size of the search space. This chapter includes recent work by the authors providing an introduction to the formal mathematical analysis of network behaviour.

1.4 METHOD

Stochastic searching is an iterative two phase process, involving the testing of cell mappings and the diffusion of successful mappings across to other cells. The general procedure of the search is;

```
INITIALISE (mappings);

REPEAT

    TEST (mappings);

    DIFFUSE (mappings);

UNTIL TERMINATION;
```

Mappings describes the [M] mapping cells which define potential mappings of the object into the search space. Each mapping cell becomes *active* or *inactive* according to the currently perceived accuracy of *its* mapping of the model in the search space.

Cell mappings are first *Initialised* to specific values. These values will usually be random unless extra information is available to cluster the mappings around a specific area.

During the *Test* phase each cell's activity changes to reflect the currently perceived accuracy of its present mapping. The accuracy of a mapping is usually defined by a local comparison between a randomly selected group of ADUs from the model and *the corresponding ADUs from the search space defined by that mapping*.

Diffusion spreads mappings from active cells to inactive cells by a stochastic selection process. Diffusion can either be synchronous, whereby all mapping updates

occur simultaneously, or asynchronous, where the cells update individually. In the networks described in this chapter all updates are synchronous.

A Stochastic network *Terminates* when its mapping cells are at equilibrium, that is, the distribution of mappings into the search space becomes constant. In some cases, for example when the ADUs in the search space are badly distorted by noise, or when the model is not within the search space, equilibrium may never be reached.

1.4.1 INITIALISATION

The initialisation of the mapping cells is determined by any a-priori knowledge of the problem that can be usefully applied.

If stochastic searching is to be used in an environment where there is no a-priori knowledge of the search space, then the *New Map* process consists of assigning all mappings randomly, (or at regular intervals through the search area). Advance knowledge of the search space enables mappings to be suitably biased. Two common examples of a-priori knowledge are;

i: The Time Ratio of the search is greater than one ([M] > [Z]).

This occurs if the number of potential mappings into the search space is less than the number of ADUs in the model. For example, classifying a model into one of [k] sets ([Z] = k), where [k] is less than the size of the model [M].

In this situation, it is certain that during initialisation at least one of the mapping cells will exhibit the correct (best fit) mapping of the model into the search space.

ii: The previous position of the model is known.

This is sometimes of use in scenarios where the model is unlikely to move in large discontinuous jumps. For example, successive images of a ball or an aircraft in flight.

In this situation the initial mappings can be biased towards the last known position of the model.

1.4.2 THE TEST PHASE

The test phase of the search evaluates the mappings currently defined by the mapping cells. The test for each mapping cell usually comprises of the selection of [j] ADUs from the model, ($1 \leq j \leq M$), which are compared with a set of [j] ADUs from the

search space, pointed to by the cell. The result of the test is either a unanimous or a majority decision of the [j] ADU comparisons. (NB. The test phase of the Hybrid Stochastic Search, described in section three, uses an n-tuple RAM neuron to make this comparison).

To avoid becoming trapped in a local minimum, at each iteration every mapping cell searches for a random section of the model. Thus when testing mapping [m] using segment [i] of the model, [j] ADUs of the model at position [i] are compared with [j] ADUs from the search space at mapping [m+i], where [i] is a random offset between [0..M-1].

If the comparison is successful the mapping cell goes *active* otherwise it remains *inactive*.

1.4.3 THE DIFFUSION PHASE

The diffusion phase enables an active cells' mapping to cross to an inactive cell via a stochastic selection procedure. Each *inactive* cells' mapping is redefined as follows;

i) Another cell is randomly selected from the network of mapping cells.

ii) If the selected cell is active then its mapping diffuses to the inactive cell.

iii) If the selected cell is also inactive then a completely new mapping into the search space is chosen using the new map function (see initialisation).

1.4.4 TERMINATION

The simplest termination condition that can be used is to sum the number of mappings that point to the same position within the search space. The algorithm terminates when this sum [Q] exceeds a given threshold. A problem with this technique is that it requires a specific threshold to be defined before use. If the model is very distorted, or there is more than one similar instance of the model in the search space, then the sum may never exceed this threshold and hence the algorithm will never terminate.

A more sophisticated termination condition is to use a stability criterion. That is, when [Q] remains constant within specified bounds, over a number of iterations, the network is said to have reached equilibrium whereby each of the [Q] cells will define the best fit mapping into the search space. However, this condition alone is not

sufficient to guarantee correct operation, since until the best fit mapping emerges and begins to diffuse across to other cells, [Q] will remain consistently low.

A general technique that has been used successfully is a combination of a threshold and stability criteria. ie. An initial activation threshold is set and the system operates until this threshold is reached. Then the system is monitored until the distribution of mappings becomes stable.

1.5 AN EXAMPLE

To illustrate the operation of a Stochastic Search a small search will be described.

The problem is to derive the position of a small symbolic pattern [123] within a search space consisting of a string of eight digits [76512329]. The model pattern can be decomposed into three symbols, ADUs (1,2,3), hence (M = 3). The search space consists of the eight possible mappings of [123] in [76512329] (given wrap-around at the boundaries), hence (Z = 8), and thus the Time Ratio for the search is (M/Z = 3/8).

The example illustrates the network converging in three iterations, (on average it would take around 2.54). Given random initialisation, at iteration one, each cell has a (1/8) probability of selecting the correct, best-fit mapping. Figure 2 shows the situation at iteration 1, one cell is testing mapping [0] with an offset of 1. Hence, the ADU pointed to in the search space is {6} which is compared with an ADU from the model pointed to by this offset, {2}. As the comparison is negative ({6}≠{2}), the cell remains *inactive*. The second cell is testing mapping [4] using an offset of 0 and thus compares {2} with {0}, hence it remains inactive, the third tests mapping [7] with an offset of 2, so comparing {6} with {3} and also stays inactive.

After this test phase, stochastic diffusion attempts to propagate successful mappings across the network. Each inactive cell selects another cell at random and if active the active cells mapping is transferred to the inactive cell. At this point none of the cells are active, thus each cell simply selects a new mapping at random into the search space.

At iteration two, one cell has selected the correct mapping, and hence becomes active. Now, during the diffusion phase the two inactive cells each have a (1/3) probability of selecting the one active cell. The probability that both will select it is (1/3 * 1/3 = 1/9), the probability that one will select it is (2 * 1/3 * 2/3 = 4/9) and the probability that neither will select it is (2/3 * 2/3 = 4/9).

At iteration three, all three cells in the network have settled on the correct mapping {3}.

SECTION TWO : ELEMENTARY ANALYSIS OF STOCHASTIC DIFFUSION

2. BACKGROUND

The complete analysis of stochastic diffusion is outside the scope of this chapter, however a theorem concerning stochastic diffusion will be proved to illustrate the general method of analysis. It is first useful to demonstrate that stochastic diffusion will converge when there is no noise.

Let the Model comprise of M ADUs.
Let *alphabet size* be the number of possible states an ADU can occupy.
Let the number of iterations be N.
A cell is *active* when its mapping points to the "correct" location of the model in the search space.

2.1 Theorem one: *Under conditions of no noise and infinite alphabet size, given one cell active at iteration one, the network is certain to converge completely, that is all mapping cells will reach the "active" state*

Let θ (K, N) be the probability that k cells are inactive at iteration n.
If at iteration n, k cells are inactive

(Insert equation [1])

(Since to remain inactive, each inactive cell must select a fellow inactive cell)

Let this quantity be called α ($\alpha < 1$ since $k < M$)

(Insert equation [2])

(Insert equation [3])

$\therefore$ Since $\alpha < 1$, θ (k, n+q) $\rightarrow 0$ as $q \rightarrow \infty$ $\forall$ k

$\therefore$ Network will converge

The analysis of stochastic diffusion uses Stochastic matrices and Markov chains. The nomenclature used is that defined by Cox & Miller[5].

2.2 Theorem two: *Under conditions of no noise and infinite alphabet size, given one mapping cell active at iteration one, the algorithm behaviour can be described by a Homogeneous Markov Chain defined by the Stochastic Matrix* **[D]**.

Let ρ (X, N) be the probability of X mapping cells active at iteration N.

If at iteration n, x mapping cells are active, and k cells are inactive (k = M - x), then

(Insert equation [4])
(Insert equation [5])

Where ρ (x, n+1) is the probability that x are active at iteration (n+1)

(Insert equation [6])
(Insert equation [7])

(Insert equation [8])
(Insert equation [9])

The above expression defines the transition probabilities, d_{ij}, of a Markov chain **[D]**, giving the probability of moving from state [i] to state [j] due to Stochastic Diffusion (see Table 1). The matrix of transition probabilities **[P]** at the nth iteration is thus defined[6] by;

(Insert equation [10])

Where P_{ij} is the probability of moving from state [i] to state [j].

All the states with [0] to [M - 1] cells on are *transient* states in the above Markov chain. These transient states represent x units being active (k = M - x). The probability of remaining in each transient state is;

(Insert equation [11])

$\therefore$ there is a (1 - α) chance that each transient state will never be entered again.

Let γ (k, β) be the probability of staying in a transient state [k] for β iterations.

(Insert equation [12])

$\therefore$ for any given transient state [k], the time periods in state [k] follow a *geometric* distribution with finite mean;

(Insert equation [13])

Given no noise, the state with [M] mapping cells active is *recurrent* and the system will remain in this state with a probability of one.

2.3 SUMMARY

The above analysis simply deals with basic stochastic diffusion, however it can be simply extended to fully describe a stochastic search by similarly considering the effects of reselection, noise and alphabet size.

SECTION THREE: THE HYBRID STOCHASTIC SEARCH NETWORK

3. THE CONNEX TEST PROBLEM V3

An example of a problem involving pattern invariance in which stochastic searching has been successfully used is that of locating eye positions within grey scale images of human faces. An adult human has little difficulty in locating eye positions from a facial image, even though these positions could be anywhere within the image field. This is a difficult task in machine vision since facial features do not generally comprise of hard edges and lines[7]. The Hybrid Stochastic Network, a combination of a Stochastic Search network and an n-tuple network, has been shown to accurately locate all eye features on which it has been trained, and also achieve over 60% success in the general problem of locating eye features on which it has not been explicitly trained.

The aims of the CONNEX project included the eventual production and demonstration of hardware Neural Network systems[7]. The Hybrid Stochastic Network comprises of two systems, both of which can be easily implemented in dedicated hardware.

3.1.1 A SYSTEM TO CLASSIFY THE DESIRED FEATURE

The n-Tuple method of pattern classification, first proposed by Bledsoe & Browning[8] and implemented in hardware as the WISARD system[9], has significant advantages over traditional methods of simple pattern classification[10]. A thorough introduction to the n-Tuple technique of pattern classification is given in [10], a brief resume of which is given below.

The data to be classified are stored as an array of boolean values. This array is then sampled, according to some predefined schema (usually either randomly or sequentially), with [n] such samples forming one n-Tuple. The data can be either exclusively sampled, such that each value is used only once, or oversampled by a factor of (m), where each value is used (m) times.

To learn a pattern, the (n) boolean values from each sample are used to form an n-bit address into an array (or RAM), where there is one such RAM per n-Tuple sample, and a boolean TRUE value is stored at this location. To classify an image, the data are sampled as before. However, instead of storing a TRUE value at each defined location, a count of the TRUE values stored at these addresses is maintained. This

count is the system response to an unknown pattern. After training with a set of patterns, such a network is able to classify patterns *not in training set*. If these new patterns are "similar" to those in the training set, then the system has *generalised* successfully. Such systems can differentiate between facial images at video speed. Hence it was decided to utilise n-tuple RAMs to perform the test phase of the Stochastic Search.

3.1.2 A SYSTEM TO SEARCH FOR THE DESIRED FEATURE

The simplest form of feature location would be to scan the classifier sequentially over the search space, however a video image of 256*256 pixels is a large space to traverse, requiring 65356 sequential comparisons (iterations). Even if a dedicated hardware system is used, working at 50 frames a second, it would still take 1300 seconds to scan the entire image! However, by using stochastic search, as described in section one, a typical analysis only requires around 30 iterations (see figure 4a), and when simulated sequentially on an *Acorn Archimedes*, takes only around one and a half minutes to locate the feature.

3.2 n-TUPLE FEATURE CLASSIFICATION

There are two particular problems with using simple n-tuple cells to classify grey level eye images: dealing with the grey level data, (simple binary coding cannot be used because of its non-linearity), and over-saturation (the network requires a lot of training in order to successfully generalise).

3.2.1 GREY LEVEL DATA

3.2.1.1 THERMOMETER CODING

An established technique to convert grey level information into binary information suitable to address the RAM neurons is Thermometer Coding (1 in N coding). This technique consists of expanding an N bit grey level into a pattern of 2^N binary pixels. Thermometer coding has the desirable property that a small change in pixel intensity causes a small change on the pattern to be classified, however it also has the undesirable property of expanding the search space exponentially with the resolution of the grey level used. This dramatically increases the amount of learning that must be carried out in order to correctly classify an object over a range of lighting conditions.

3.2.1.2 MINCHINTON CELLS

Minchinton cells describe a family of analogue preprocessing units that can be successfully used with n-tuple nets[11]. In the simplest case they consist of a digital comparator which is used to sample two randomly chosen points of the input pattern, the cell firing if ($P_i > P_j$). It can be shown that;

a: If the input pattern is sufficiently oversampled, non of the grey level information is lost.

b: The output of the cell is immune to global DC changes in the input pattern.

Hence Minchinton cells were used to convert the grey level data into binary data.

3.2.2 SATURATION

The stochastic search technique ideally requires a very sharp response from the feature classification system as the time to converge in an Stochastic Search increases rapidly as the response of the classifier to non-features rises. This would nominally require the tuple size to be large or the amount of learning small. However, both these actions reduce the ability of the n-tuple network to generalise.

3.2.2.1 ADAPTIVE n-TUPLE NETWORKS

To improve on the standard n-tuple behaviour a system for controlling the saturation in each RAM was used. This system differs from the standard single layer n-tuple learning in that, instead of storing a BOOLEAN value at the address defined by the micro feature of the image covered by the RAM, a count is maintained of how frequently that micro-pattern has been observed during training. At the end of the learning period only the addresses containing the highest (l) counts will cause the given RAM to fire. Thus by control of (l) there is a mechanism for effectively controlling saturation in each RAM.

3.3 NETWORK ARCHITECTURE

Figure 3a shows a schematic block diagram of the Hybrid Stochastic Network. The input video image is first digitised to a resolution of 256*256 pixels at an accuracy of 8 bits. A selection of points from this grey level data is processed by an array of Minchinton cells (figure 3b), the selection being controlled by the stochastic mapping cells. Each n-tuple RAM thus receives binary data from a given mapping into the search space. The response of the RAM to this data comprises the Test phase of the stochastic search. The diffusion phase then reallocates stochastic cell mappings dependent on their success.

3.4 EXPERIMENTS

Due to hardware limitations it was unfortunately not possible to use the official CONNEX test data, as it was not in a form readable by the Archimedes computer on which the system was developed. However, by filming a set of subjects in conditions similar to those described in the CONNEX test document[7], a close approximation to real CONNEX test data was obtained.

The test data used comprised of 64 video images of forward facing human faces, digitised to a resolution of 256*256 pixels at an accuracy of eight bits. The images were vaguely normalised to the centre of the screen. The subjects chosen were both men and women of various race. There were no restrictions on their clothing, beards or make-up.

3.4.1 LOCATION OF 'TRAINED' FACIAL FEATURES

This test was a simple preliminary evaluation of the networks ability to locate a facial feature on which it had been trained. This involved training the network on one set of eyes then testing the system on a different image of the same face.

The system correctly located both eyes on each occasion it was tested.

3.4.2 LOCATION OF 'UNTRAINED' FACIAL FEATURES

The second experiment tested how well the Hybrid Network located eyes over which it had not been previously trained. This involved training the net on 32 subjects (64 eyes) with their pupils approximately central, and testing the system using a different set of 32 subjects.

This experiment was repeated three times. Each training session lasted approximately three hours and each test session five hours. The tuple sizes used were four, with a saturation level of two (ie. only two addresses in each RAM would cause the cell to fire); and eight, using saturation levels of eight and ten. The results are summarised in figure four (t is the tuple size and l is the degree of saturation). It can be observed from figure 4a that, with a tuple size of eight ($t = 8$) the network usually converged within around 30 iterations. This figure also illustrates how convergence is reached more rapidly with a high tuple size and low saturation ($t = 8, l = 8$). This is due to the stochastic search operating more efficiently with a sharp filter response. However figure 4b shows a converse effect, as the largest number of correct convergences occur when the filter is relatively blurred ($t = 8, l = 10$). This is due to the n-tuple cells generalising more efficiently over the untrained eye images with the higher RAM saturation.

3.5 SUMMARY

The investigation of the Hybrid Stochastic Network as part of the CONNEX project demonstrated the feasibility of using such an architecture for feature location, however more research needs to be carried out to improve the performance of the system, and to investigate its behaviour in more general situations. The network developed was able to locate eye features in over 60% of cases tested. The current system used no explicit knowledge of where the facial features were likely to be in the image.

The performance of the Hybrid Stochastic Network described was limited not by the stochastic search, but by the n-tuple classification sub-system. Overall system performance could be substantially improved if the classifier had a sharper rejection of non features but better generalisation of untrained correct features.

This research was carried out as part of the British Telecom CONNEX project.

4. REFERENCES

[1] Hinton, G.E. (1981)
"*A Parallel Computation that assigns canonical object based frames of reference*", Proc. 7th Int. Jnt. Conf. A.I.

[2] McClelland, J.L. (1981)
"*Retrieving General and Specific Knowledge From Stored Knowledge of Specifics*", Proc. 3rd Conf. Cognitive Society, Berkeley, CA.

[3] Bishop, J.M. (1989)
"*Stochastic Searching Networks*", Proc. 1st IEE Conf. on Artificial Neural Networks, pp. 329-331, London.

[4] Bishop, J.M. (1989)
"*Anarchic Techniques for Pattern Classification*", pp. 5.14, PhD Thesis, Reading University

[5] Cox, D.R. & Miller, H.D. (1965)
"*The Theory of Stochastic Processes*", Chapter 3, Chapman & Hall, London

[6] Cox, D.R. & Miller, H.D. (1965)
"*The Theory of Stochastic Processes*", pp. 79, Chapman & Hall, London

[7] Welsh, W.J., Woodland, P.C., & Myers, D.J. (1989)
"*A Set of Test Problems for Assessing Neural Net Algorithms*", BTRL, Martlesham Heath, UK.

[8] Bledsoe, W.W & Browning, I. (1959)
"*Pattern Recognition and Reading by Machine*", Proceedings of the Eastern Joint Computer Conference. pp. 225-232.

[9] Aleksander, I., Thomas, W.V., & Bowden, P.A. (1984)
"*WISARD a Radical Step Forward in Image Recognition*", Sensor Review, July 1984.

[10] Aleksander, I & Stonham, T.J. (1979)
"*Guide to Pattern Recognition using Random Access Memories*", Computers & Digital Techniques, vol.2 No. 1, pp. 29-40

[11] Bishop, J.M., Minchinton, P.R. & Mitchell, R.J. (1990)
"*The Minchinton Cell - Analogue Input to the n-tuple Net*", Proc. INNC '90. Paris.