

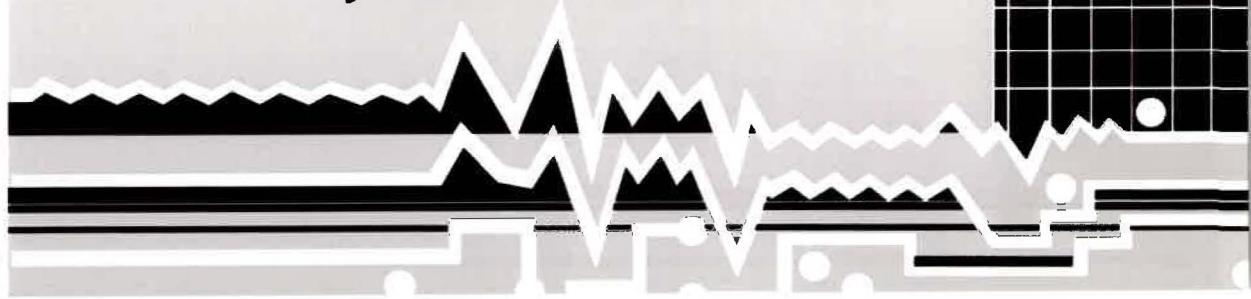
AUSTRALIAN
JOURNAL

OF

INTELLIGENT
INFORMATION
PROCESSING
SYSTEMS

Volume 1, No. 2

June 1994



Editorial Policy

The Australian Journal of Intelligent Information Processing Systems is an interdisciplinary forum for providing latest information on research developments and related activities in the design and implementation of intelligent information processing systems.

The areas of interest include, but are not limited to:

- artificial intelligence
- artificial neural networks
- computer science
- fuzzy systems
- virtual reality

The journal publishes both theoretical and application-oriented papers in diverse areas related to intelligent systems. The journal also publishes survey articles, short notes, Ph.D. thesis abstracts and project reports with emphasis on Australian projects.

Editorial Board

Yianni Attikiouzel

Editor in Chief

*The University of Western Australia,
Australia*

James C. Bezdek

University of West Florida, USA

Tony Constantinides

*Imperial College of Science, Technology &
Medicine, U.K.*

John Debenham

University of Technology Sydney, Australia

Tom Downs

The University of Queensland, Australia

Peter Eklund

University of Adelaide, Australia

Norman Foo

Sydney University, Australia

Toshio Fukuda

Nagoya University, Japan

Ray Jarvis

Monash University, Australia

Dorota Kieronska

Curtin University of Technology, Australia

Andrew Lister

University of Queensland, Australia

Robert Marks

University of Washington, USA

D. Nandagopal

*Defence Science & Technology Organisation,
Australia*

Marimuthu Palaniswami

The University of Melbourne, Australia

Patrick Simpson

Orinconc Corporation, USA

M.V. Srinivasan

Australian National University, Australia

A. C. Tsoi

The University of Queensland, Australia

A. Venetsanopoulos

University of Toronto, Canada

K. P. Wong

*The University of Western Australia,
Australia*

Responsibility for the content of these papers rests upon the authors and not with the publishers. Data and conclusions developed by the authors are for information only and are not intended for use without independent substantiating investigation of the part of the potential user.

The Australian Journal of Intelligent Information Processing Systems is published quarterly. All correspondence, including manuscripts and advertising enquiries should be addressed to the Editor in Chief. Abstracting is permitted with due credit to the Australian Journal of Intelligent Information Processing Systems. Individual rate: \$60 within Australia, \$70 overseas; institutional rate: \$200 within Australia, \$230 overseas. Individual copies can be purchased at \$15 per single copy. REPRINTS: reprints of technical articles are available, quantities of no less than 50 can be ordered.

Australian Journal of Intelligent Information Processing Systems

ISSN: 1321-2133

Volume 1, No. 2

June 1994

CONTENTS

ARTIFICIAL NEURAL NETWORKS

On Two Methods of Accelerating Learning in Feedforward Networks <i>R.J. Gaynier, T. Downs</i>	1
Efficient Multichannel Image Recognition Using Feedforward Multiresolution Neural Networks <i>D. Kalgeras, S. Kollias</i>	8
A Back-Propagation Adaptation Yielding a Ternary Output ANN with Limited Integer Weights <i>G.R. Cukowicz, M.R. Haskard</i>	16

ARTIFICIAL INTELLIGENCE

A Comprehensive Synthesis Strategy for Conflict Resolutions in Distributed Expert Systems <i>M. Zhang, C. Zhang</i>	21
Family Law, Natural Language Understanding and AI <i>W.K. Yeap</i>	29

GENETIC ALGORITHMS

An Introduction to Evolutionary Computation <i>D.B. Fogel</i>	34
--	----

IMAGE PROCESSING

A Modified Lagrange's Interpolation for Image Reconstruction <i>S. Suthaharan</i>	43
--	----

SELECTED AUTHOR BIOGRAPHIES

ABSTRACTS OF THE AUSTRALIAN UNIVERSITIES POSTGRADUATE RESEARCH THESES

CALENDAR

Dear colleague

It is my pleasure to welcome you to another issue of the Australian Journal of Intelligent Information Processing Systems - AJIIPS.

As with any new journal no matter how important and timely it is, the first few issues are difficult to produce without financial support. I am grateful to the Department of Commerce and Trade of the Government of Western Australia that has sponsored the printing and mailing of this issue. Present subscribers to AJIIPS will get an extra two free issues. I would like to take this opportunity to encourage you to subscribe to the Journal personally or request your library to subscribe.

As you can see from the contents of this issue, technical papers from around the world will be welcome. Nevertheless, sections discussing Australian developments, including postgraduate theses, abstracts and project reports, will be permanent features of AJIIPS. Information about forthcoming events - conferences, exhibitions and other events of interest to our audience, will also be included.

The quality of the journal is dependent on the quality of submitted work. I would like to take this opportunity to invite researchers and practitioners working in any of the areas related to intelligent systems to submit their work for publication, either as technical papers or as reports of projects. Our aim is to publish a journal that will bring together researchers from engineering, computer science and other related areas. This can be achieved by contributions to a journal that is widely read by a multi disciplinary section of our research community and industry.

There are a number of paper categories that will be accepted:

AJIIPS will publish theoretical, research and application oriented papers, including but not limited to the following areas:

- | | |
|---|---|
| <ul style="list-style-type: none">• Artificial Intelligence• Knowledge Based Systems• Fuzzy Logic• Genetic Algorithm• Distributed Systems | <ul style="list-style-type: none">• Neural Networks• Parallel Processing• Virtual Reality• Evolutionary Computation• Applications |
|---|---|

There are a number of paper categories that will be accepted:

- | | |
|---|--|
| <ul style="list-style-type: none">• Full papers• Abstracts of postgraduate theses• Conference reports• Research Centre reports | <ul style="list-style-type: none">• Short communications• Calendar of events• Book reviews• Project reports |
|---|--|

Finally, I would like to express my sincere thanks to the WA Department of Commerce and Trade for their generous support.

Yianni Attikiouzel

On Two Methods of Accelerating Learning In Feedforward Networks

R.J. Gaynier, T. Downs

Centre for Artificial Neural Networks and their Applications

Department of Electrical and Computer Engineering

The University of Queensland

Brisbane Q 4072, Australia

Abstract

It is well known that the backpropagation algorithm is usually very slow to converge to a solution and this has prompted researchers to seek improved methods of training feedforward networks. This paper describes novel implementations of two approaches to the acceleration of learning speed that have proved beneficial in the past. These two approaches are (i) the use of error functions other than mean-squared error and (ii) the introduction of noise into the learning process. In this paper a new error function is introduced which is based upon the properties of error signals in the backpropagation process. The error function is shown to provide very substantial increases in learning speed for problems with dynamic exceptions and is also shown to have the ability to correct misclassified patterns when used in an adaptive fashion. The paper also shows how to introduce noise into learning in a controlled fashion, so that higher levels of noise are introduced where they are most needed. Examples are given to demonstrate that this approach is also capable of providing substantial increases in learning speed.

1 Introduction

The most widely used method of training artificial neural networks is the backpropagation algorithm. The method has achieved some spectacular successes in the last decade or so and this is in spite of the fact that the algorithm is usually very slow to converge for problems of practical interest. This slowness has prompted many researchers to seek improved versions of the backpropagation algorithm.

Backpropagation is a gradient descent algorithm that seeks to minimize mean-squared error. Variations on the algorithm that have been proposed in order to increase speed of convergence include the use of an adaptive learning rate [1], implementation of line search [2], the use of alternative error functions [3], [4] and the introduction of noise into the learning process [5], [6]. This paper is concerned with the latter two variations on backpropagation.

In Section 2 we discuss the properties of error functions that affect the speed of backpropagation learning. On the basis of this discussion, we propose a new error function and demonstrate how appropriate shaping of the function can lead to radical improvements in learning speed in comparison to the speeds available from other proposed error functions. We go on to demonstrate how, by allowing a parameter of our error function to vary adaptively during the learning process, a feedforward neural network can be provided with the ability to give a correct classification for patterns that have been misclassified in the training data.

In Section 3 we investigate two novel methods of

introducing noise into the backpropagation learning process. One of these is concerned with learning in networks of Heaviside units for which a backpropagation-like algorithm is described in [7]. The second method applies to networks of sigmoidal units. Both methods allow noise to be introduced in an automatically-controlled fashion and both are shown to provide substantial improvements in learning speed.

Section 4 summarises the developments presented in the paper and points out areas that could benefit from additional study.

2 A New Error Function for Backpropagation Learning

In the last few years several alternatives to the mean-squared error function have been proposed as a means of improving the speed of backpropagation learning. Perhaps best-known among these are the functions considered in [3] and the cross-entropy function discussed in [4]. In this section we investigate properties of error functions that influence the rate of learning when backpropagation is employed.

In backpropagation learning it frequently happens that a substantial proportion of the training set is learnt very quickly, but the remaining patterns take much longer to be learnt, and in some cases may not be learnt at all. This behavior can occur for a number of reasons, two of them being:

- (i) static exceptions in the training data, which can

be due to errors in the training data (e.g. misclassified or noise-affected patterns) or to training data that are abnormal or contradictory in some way;

- (ii) **dynamic exceptions**, which can arise when the training data contain no static exceptions; they occur when a network's weight values make it difficult to accommodate those patterns not yet learnt, causing them to appear to the network as exceptions.

A recent paper [8] identified three particular properties of error functions that tend to cause dynamic exceptions to occur and in the following we propose a new error function that allows these properties to be investigated rather more closely than was attempted in [8]. These investigations allow us to draw conclusions about the desirable shape of the error function used in backpropagation learning and we show that, when our error function is appropriately shaped, it provides very substantial increases in learning speed over existing functions for problems with dynamic exceptions. We then go on to show how the shape of our error function can be made variable during learning in order to achieve good performance on problems containing static exceptions.

2.1 Error Signal Properties Affecting Learning

In backpropagation learning, an error signal is propagated backwards from the output units and is used to update network weights. The error signal is equal to the product of the derivative of the error function and the derivative of the activation function of the units in the network. In the process of learning a set of patterns, the error signal for the full set is equal to the sum of the individual contributions produced by each pattern. And clearly, if the contribution made to the error signal by poorly-learned patterns is not sufficiently large, its effect will be overridden by the combined effect on the error signal of the better-learned patterns. In [8], three different properties of error signals that can cause this to occur were identified:

- (i) The error signal may approach zero too slowly for small error values so that the contributions to the error signal from well-learned patterns will be large.
- (ii) The error signal from poorly-learned patterns may never reach a large enough value to have any effect on the learning process (i.e. the maximum value of the error signal may be too small).
- (iii) Large errors drive activations toward saturation values in the sigmoidal characteristic causing the derivative of the activation function, and hence the error signal, to tend to zero. Thus very poorly-learned patterns contribute little to the

overall error signal.

Having identified these three properties, the authors of [8] then proposed a new error function which they argued should perform well on problems involving dynamic exceptions. Using this error function, which they referred to as the Exception Error Function (EEF), they demonstrated good learning performance on N-2-N encoder problems (which are known to be prone to dynamic exceptions when standard backpropagation is applied [9]).

We have investigated at some length the influence on learning speed of each of the above properties, and this was achieved by means of the error function

$$E = \frac{\alpha(x^2+x^4)}{1-x^2+\beta x^4} \quad (1)$$

where x is the error at the network output and α, β are parameters that allow the shape of E (and, more particularly, the shape of the error signal) to be varied. We refer to E as the Test Error Function (TEF) and its shape for $\alpha = 1$ and a range of values of β is shown in fig 1(a). Note that although α is merely a scale factor in E , it can be used to vary the value of the maximum of the error signal and so has a useful role to play. The error signals corresponding to the error functions in fig 1(a) are shown in fig 1(b), where the error signal for the EEF is included for comparison purposes¹.

Our investigations of the effects of different shapes of error signal on learning speed led us to conclude that an additional property of the error signal (not considered in [8]) appears to have the greatest influence. This property is the location of the maximum of the error signal - note that in fig 1(b) the maximum shifts to the right with decreasing β . Our investigations also indicate that if the maximum of the error signal is too large it has a detrimental effect on learning. The overall effects of the shape of the error signal are discussed below.

2.2 Results

Table 1 presents our results for the 14-2-14 and 19-2-19 encoder problems using what we believe is an approximately optimal shape for the error signal of the TEF. These results indicate a clear superiority of the TEF over the other three error functions for N-2-N encoder problems. The reasons for the superiority of the TEF over the EEF are, in our view, evident from fig.1(b). The fact that the error signal for the EEF tends to infinity for large error and that it is also large for "moderate" error indicates to us that the EEF places too much emphasis on exceptional patterns to the detriment of learned, or nearly learned, patterns. In contrast, the error signal for the TEF is small for low and moderate error values and then reaches a *finite* maximum near maximum error.

¹The activation function employed was $1/(1+e^{-x})$.

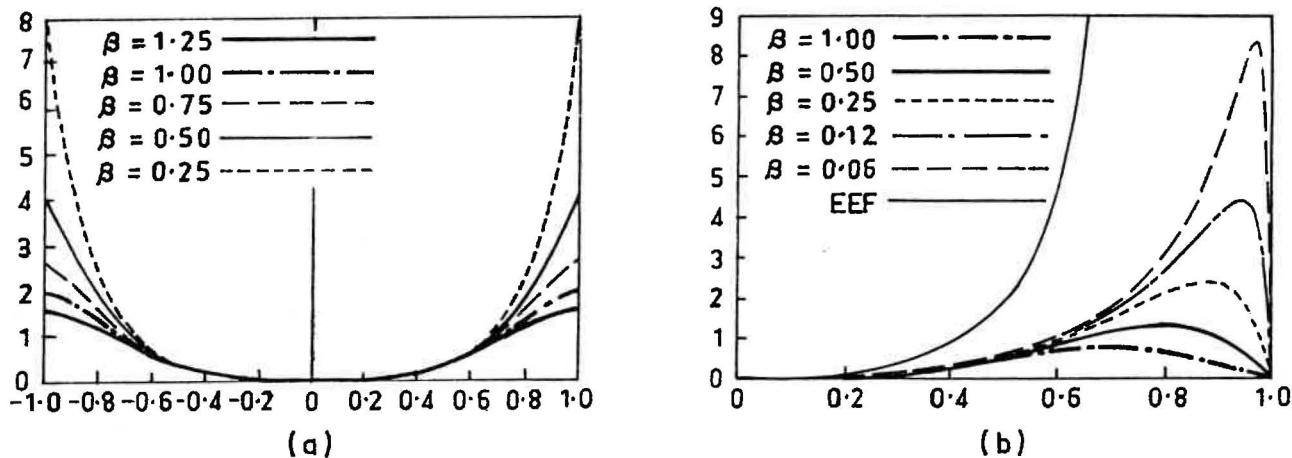


Figure 1: (a) TEF for various β with $\alpha = 1$; (b) error signals of the TEF (the error signal of the EEF, which goes rapidly to infinity, is also shown).

14-2-14 Encoder					19-2-19 Encoder			
Func.	Least	Median	Most	Conv	Least	Median	Most	Conv
TEF	575	1606	2021	10/10	8178	12414	23524	10/10
EEF	2234	4397	5175	10/10	59039	97274	161484	8/10
X-Ent	126696	169554	405960	10/10	-	-	-	0/10
M Sq	221102	441074	735487	10/10	-	-	-	0/10

Table 1 : N-2-N encoder for the TEF, EEF, Cross Entropy and Mean Squared error functions. Results for the latter three from [8].

Training Errors					Test Errors		
Func.	Epochs	Least	Median	Most	Least	Median	Most
VEF	500	6	10	12	0	1.5	4
X-Ent	500	4	7	8	2	3	6
EEF	500	7	14	31	4	5	19
TEF	500	15	20	57	6	13	28
Mean Sq	500	386	459	638	386	459	638

Table 2 : Results for the Contiguity problem with 10 static exceptions

Given our success with a problem that involves dynamic exceptions, we decided to investigate the possibility of using a similar approach to improve learning performance for problems involving static exceptions. To this end, we chose to consider the contiguity (2 or more clumps) problem which was used in [4] as a benchmark problem for the cross-entropy function. Static exceptions can be introduced into this problem by simply misclassifying

some of the patterns.

In the case of the N-2-N encoder, our prime concern was learning speed, but in a case where the training data contains misclassified patterns, our main interest is in how these patterns are dealt with by the learning process. And, in the latter case, it is generally desirable that a network should fail to learn most (preferably all) of the

misclassified patterns because otherwise its generalisation capabilities are likely to be impaired. For this to be achieved, we require that the large error at the output due to a misclassified pattern lead to only a small error signal. Thus, in order to deal with static exceptions in this way, an error signal like that of the EEF in fig.1(b) is out of the question. The other error signals in fig.1(b) do, however, offer the possibility of suitable treatment of static exceptions.

But in attempting to deal satisfactorily with static exceptions, we must not forget that dynamic exceptions have to be accommodated also. Our approach is based on the assumption that dynamic exceptions will gradually be accommodated so long as the error is large for large error in the earlier stages of training. We commence with an error signal like the one with the maximum peak ($\beta = .06$) in fig.1(b) and, over time, gradually increase β so that the peak decreases and shifts to the left. In this way we seek to accommodate the dynamic exceptions (whose error should gradually reduce) and exclude the static exceptions (whose error will remain large). We refer to an error function that changes in this way during the learning process as a VEF (Variable Error Function).

Table 2 shows the results we obtained on a ten-input, ten-hidden unit, single-output network given the task of learning the full set of 1024 patterns in the 10-bit contiguity problem. In this table, the data entitled "Training Errors" relate to the training of the network with 10 patterns misclassified; the data entitled "Test Errors" relate to the testing of the trained network on the complete contiguity problem with no misclassifications. It is interesting to note that the VEF tends to reject the 10 misclassified patterns during training and, as a consequence, performs best on the test set. Note that in rejecting the 10 misclassified patterns the VEF is effectively correcting the misclassifications.

3 Two Controlled Methods of Introducing Noise Into The Learning Process

It has frequently been reported that significant improvements in the performance of training algorithms for neural networks can be achieved by adding noise to the training process (see, for example, [5], [6]). But in the techniques that have so far been described, noise is generally introduced on a trial-and-error basis with the most suitable amount of noise being determined empirically. Too much noise has a deleterious effect on the learning process, and too little has little or no effect.

We have recently developed a training algorithm for a class of networks that provides an analytical framework for the automatic control of the amounts of noise introduced and which, unlike existing techniques, allows provision of differing amounts of noise (as required) to different parts of the network. The algorithm that provides the analytical

framework was described in [7] and was developed for the training of networks whose neurons have Heaviside characteristics. In this algorithm, the activations of the network's hidden units are treated as continuous random variables and this fact allows direct extension to controlled noise introduction.

In this section we first briefly describe our algorithm for training networks of Heavisides and then show how it can be adapted for the introduction of noise. We give some results that indicate a significant improvement in learning performance. We then show how a similar approach can be applied to the training of networks of sigmoids and again demonstrate substantial performance improvements.

3.1 An Algorithm For Training Networks of Heavisides

In common with several other training algorithms [10-12] our method for training networks of Heavisides [7] incorporates internal representations into the learning process. In [10], [11] the internal representations are the output values of the hidden units, but in our algorithm in [7] the internal representations we use are the activations of the hidden units as in [12]. We treat these activations as continuous random variables. This allows us to express the outputs of the hidden units in probabilistic form and provides us with a cost function that is a differentiable function of the means and variances of the internal representations. And hence the means and variances of the internal representations can be adjusted to reduce the cost function and therefore improve the internal representations. Once suitable internal representations have been found, the search for suitable weights reduces to a set of single layer problems. Training is a two step process, the first step being the adjustment of the weights while holding the internal representations constant, and the second step being the adjustment of the internal representations while holding the weights constant. Training alternates between these two steps, making small adjustments in the weights and internal representations, until the network converges to a solution.

For clarity, let us initially consider a network with a single layer of hidden units. The algorithm begins with randomly chosen values of mean and variance for the internal representations (recall that these are the activations of the hidden units) and randomly chosen values for the network weights. The first step is to adapt the weights in an effort to find mappings from input to internal representation and from internal representation to output. This is a single layer learning problem and delta rule learning is applied. During this step the activations of the hidden units are set equal to the current mean values of the internal representations. Each layer is treated independently during training. The internal representations are used as inputs during training of the output layer and as targets during the training of the input layer. The changes made to the weights at this step are as follows :

$$\Delta w_{ji} = \{T_{jt} - H[w_{ji}H(M_{it})]\}H'(M_{it}) \quad (2)$$

$$\Delta w_{ik} = (H(M_{it}) - y_{it})I_{kt} \quad (3)$$

where w_{ji} is the weight connecting output unit j to hidden unit i and w_{ik} is the weight connecting hidden unit i to input k . T_{jt} and y_{jt} are the target and actual outputs of unit j for input pattern t . I_{kt} is the value of input k for pattern t . M_{it} is the mean of the internal representation for unit i and input pattern t and H is the Heaviside transfer function.

In the second step the mean and variance of the internal representations are adjusted using ideas similar to those underlying backpropagation. First note that we can calculate the probabilities of the hidden units attaining particular values (given the mean and variance of the internal representations) as follows :

$$Pr\{y_{it} = 1\} = \Phi\left(\frac{M_{it}}{S_{it}}\right) \approx G\left(\frac{M_{it}}{S_{it}}\right) \quad (4)$$

where Φ is the cdf of the standard normal distribution given mean M_{it} and variance S_{it} of the internal representation. The function G is an analytic approximation to the standard normal² used so that we can differentiate the probabilities with respect to M_{it} and S_{it} . We now define the cost function

$$E = \frac{1}{2} \sum_t \sum_j \sum_i [(w_{ji}(Pr\{y_{it} = 1\} - Pr\{y_{it} = -1\}) - T_{jt})^2] \quad (5)$$

This cost function describes the mean square error between the target outputs and the outputs that would be produced if the activations of the hidden units were equal to the internal representations.

The analytic approximation of equation (4) can now be substituted in equation (5) and the derivatives required for a gradient descent algorithm (i.e. $\partial E / \partial M_{it}$ and $\partial E / \partial S_{it}$) can be readily calculated. In implementing this algorithm, we employ certain refinements that improve the rate of convergence; details of these refinements are given in [7].

3.2 Modifying the Algorithm to Incorporate Noise

For any given learning situation there is generally an optimum amount of noise that should be introduced, but this is not known *a priori*. In our algorithm, the fact that we are treating internal representations as continuous random variables provides us with a framework for introducing noise in amounts that are automatically

controlled. Recall that each iteration of the algorithm described in the previous section has two steps. In one step, the means and variances of the internal representations are adjusted so as to reduce the cost function. In the other step, the values of the network weights are adjusted, and in this step the internal representations are treated as deterministic variables with their values set equal to the mean values that have just been computed.

Noise can be very simply introduced (in a controlled fashion) into this process by modifying slightly the manner in which the network weights are adjusted. Instead of setting the values of the internal representations equal to their mean values, we can, for each internal representation, employ a sample from the (normal) distribution that is defined by its current values of mean and variance. This sampling process imposes noise on the weight adjustment step of our algorithm, but the degree of noise imposed is controlled by the other step, which determines the variance of each internal representation.

This approach to the addition of noise to the learning process appears to be novel. Unlike other methods, different variables are subjected to differing amounts of noise. The amounts vary depending upon how effectively the network's internal representations are adapting to the learning problem. In general (and in our computational experience) internal representations that are "good" have low variances and those that are "bad" have high variances. So, effectively, larger noise is added in those parts of the network where improvements are needed.

The new version of our algorithm was compared with the earlier version for three problems : (i) the XOR problem with 2 hidden units, (ii) the 3-bit parity problem with 4 hidden units, and (iii) the 8-4-8 encoder/decoder. The results obtained are shown in Table 3. This table indicates a very significant increase in the percentage of successful training attempts achieved by the new version of our algorithm. The table also shows how the average number of iterations required to achieve success is substantially reduced when using the new version.

3.3 Applying a Similar Approach to Networks of Sigmoids

Given our success in improving learning performance for networks of Heavisides, it is natural to ask whether it is possible to apply a similar technique to networks whose neurons have sigmoidal characteristics. The answer is in the affirmative, although the procedure we have so far developed differs from the one described earlier in that it is not governed by a fully analytic framework. In the Heaviside case, the amount of noise introduced at each node is governed by a normal distribution whose mean and variance are automatically determined by the learning algorithm. For the sigmoid case, we have produced an algorithm which is similar in spirit to our Heaviside algorithm, but which relies upon a predetermined function

²The approximation we employed was $G(x) = \frac{1}{2}(1 + \tanh(\frac{x}{\sqrt{2}}))$ where $y = 0.7988x(1 + 0.04417x^2)$. According to [13], the maximum absolute error incurred in using this approximation is 0.000140.

	XOR	Epochs	3 BIT PARITY	Epochs	8 4 8 ENC/DEC	Epochs
STANDARD	50%	477	84%	419	68%	1557
RANDOM	94%	234	98%	382	74%	1038

Table 3 : Percentage success rates and average number of epochs. "STANDARD" used the algorithm in [7] and "RANDOM" used random samples of the internal representations given their means and variances for the adaptation of the weights.

	Training Accuracy	Test Accuracy	Hidden Units	Epochs
BACKPROP	100%	100%	3	390
RANDOM	100%	100%	3	98

Table 4 : Results for the MONK's Problem 1, training and test accuracy and average number of epochs.

	Training Accuracy	Test Accuracy	Hidden Units	Epochs
BACKPROP	100%	100%	2	90
RANDOM	100%	100%	2	54

Table 5 : Results for the MONK's Problem 2, training and test accuracy and average number of epochs.

	Training Accuracy	Test Accuracy	Hidden Units	Epochs
BACKPROP	-*	93.1%	4	190
RANDOM	100%	93.1%	4	152

Table 6 : Results for the MONK's Problem 3, training and test accuracy and average number of epochs.

* no data available from [14].

for establishing the required noise variance at each node.

As in the Heaviside case, noise is introduced on the internal representations but, because we are now dealing with sigmoidal neurons, we now use the outputs of the hidden units (rather than their activations) as the internal representations. Other than the mechanism we use for noise introduction, a standard backpropagation algorithm is employed. The amount of noise introduced at each node is a function of the backpropagated error and, for the results to be presented here, was determined by the formula

$$S_{it} = 1 - \frac{2.5e^{-x^2/2}}{\sqrt{2\pi}} \quad (6)$$

where S_{it} is the variance of the internal representation at hidden unit i for training example t and x is the

backpropagated error at hidden unit i . This function is an "inverted" normal density curve so that as the backpropagated error decreases or increases the variance decreases or increases but is bounded. Using this equation we update the variance after each training iteration.

To test how well this procedure works we tested it on the MONK's problems [14]. The MONK's problems are three binary classification tasks. "Robots" are described by six different attributes and the task of each problem is for the network to learn to dichotomise the robot population given a subset of all possible examples as a training set. For example, in the first MONK's problem robots with the attributes (head shape = body shape) OR (jacket color is red) belong to one classification and all the remaining robots belong to the other classification.

The results for our algorithm presented in Tables 4, 5, and

6 are averaged over ten attempts, 1000 maximum data presentations per attempt, for each of the three MONK's problems. The results for backpropagation with noise are taken directly from [14]³. The tables indicate a clear reduction in the number of epochs required for training when noise is added to simple backpropagation in the manner described.

4 Conclusion

In this paper we have investigated two approaches to the acceleration of learning in feedforward networks. We have studied the properties of backpropagation error signals that affect learning speed by means of a new error function and shown that appropriate choice of the parameters of this function leads to a radical increase in learning speed for problems with dynamic exceptions. We have also shown that by allowing one parameter of the error function to vary during the learning process, it is possible for a neural network to correct input data involving erroneously classified patterns. In the case of other types of static exception, the network would reject some of the exceptional patterns. This behaviour appears worthy of further investigation.

We have also described a method of introducing and controlling noise during training and this method has several advantages over existing techniques. It allows variable amounts of noise to be introduced at each network node with each noise source being separately controlled. In the case of Heaviside networks the amounts of noise introduced are constrained within an analytical framework while in sigmoidal networks the control is somewhat more *ad hoc*, although it is constrained by the backpropagated error. In both cases, substantial increases in learning speed were demonstrated. Further study will be required in order to determine the optimum (or near-optimum) procedure to apply in the sigmoidal case.

5 References

- [1] Jacobs, R.A., "Increased rates of convergence through learning rate adaptation", *Neural Networks*, Vol. 1, pp. 295-307, 1988.
- [2] Hush, D.R. and Salas, J.M., "Improving the learning rate of backpropagation with the gradient reuse algorithm", *Proceedings IEEE International Conference on Neural Networks*, San Diego, 1988, Vol. I, pp. 441-447.
- [3] Fahlman, S.E., "Fast-learning variations on backpropagation : an empirical study", *Proceedings 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton and T. Sejnowski (eds), San Mateo, Morgan Kaufmann, pp. 38-51, 1989.
- [4] Solla, S.A., Levin, E. and Fleisher, M., "Accelerated learning in layered neural networks", *Complex Systems*, Vol. 2, pp. 625-639, 1988.
- [5] Von Lehmen, A., Paek, E.G., Liao, P.F., Marrakchi, A. and Patel, J.S., "Factors influencing learning by backpropagation", *Proceedings IEEE International Conference on Neural Networks*, San Diego, 1988, Vol. I, pp. 335-341.
- [6] Murray, A.F., "Multilayer perceptron learning optimized for on-chip implementation : a noise-robust system", *Neural Computation*, Vol. 4, pp. 366-381, 1992.
- [7] Gaynier, R.J. and Downs, T., "A method of training multi-layer networks with Heaviside characteristics using internal representations", *Proceedings IEEE International Conference on Neural Networks*, San Francisco, 1993, Vol. III, pp. 1812-1817.
- [8] Lister R., Bakker, P. and Wiles, J., "Error signals, exceptions and backpropagation", *Proceedings International Joint Conference on Neural Networks*, Nagoya, 1993, pp. 573-576.
- [9] Lister, R., "Visualizing weight dynamics in the N-2-N encoder", *Proceedings IEEE International Conference on Neural Networks*, San Francisco, 1993, Vol. II, pp. 684-689.
- [10] Krogh, A., Thorbergsson, G.I., Hertz, J.A., "A Cost Function for Internal Representations", *Advances in Neural Information Processing Systems 2*, pp. 733-740, Morgan Kaufmann, 1990.
- [11] Grossman, T., "The CHIR Algorithm for Feed Forward Networks With Binary Weights", *Advances in Neural Information Processing Systems 2*, pp. 516-523, Morgan Kaufmann, 1990.
- [12] Rohwer, R., "The 'Moving Targets' Training Algorithm", *Advances in Neural Information Processing Systems 2*, pp. 558-565, Morgan Kaufmann, 1990.
- [13] Kotz, S., Johnson, N.L., (eds.), *Encyclopaedia of Statistical Sciences*, Wiley, New York, 1985.
- [14] Trun, S.B., et. al., "The MONK's Problems, A Performance Comparison of Different Learning Algorithms", Carnegie Mellon University, CMU-CS-91-197, December 1991.

³We ourselves have obtained very similar results for these problems using backpropagation without noise

Efficient Multichannel Image Recognition Using Feedforward Multiresolution Neural Networks

Dimitrios Kalogerias and Stefanos Kollias
Computer Science Division
National Technical University of Athens
Zografou 15773, Athens, Greece
tel (30)-1-7757401, fax (30)-1-7784578
e-mail: dkalo@ntua.gr

Abstract

An innovative approach is proposed in this paper for recognition and interpretation of multichannel images, containing various types of information, such as 2-D gray scale, color and 3-D range data. The method includes multiresolution analysis using wavelet filter banks to obtain hierarchical representations of the images, and applies two types of feedforward neural network classifiers, multilayer perceptrons and probabilistic neural networks, to classify the images at each resolution level. A novel constructive technique for the efficient training of the networks at various resolutions is described, while simulation results using real image data are presented, illustrating the performance of the method.

Keywords: multiresolution analysis, multilayer perceptron, probabilistic network, multichannel image recognition.

1 Introduction

Existing industrial inspection and recognition systems generally store a number of different models and geometrical features and use a pattern matching scheme, based on these features, for classification and recognition of acquired 2-D gray-scale images of objects or scenes [19, 4]. A problem that is presented, if classification of 3-D range or color information is included in the capabilities of such systems, is that the amount of information that has to be processed by the system can be extremely high. An innovative multiresolution image decomposition and classification/recognition scheme is proposed in this paper for handling the resulting information based on neural network architectures.

Artificial neural networks have been shown capable of extracting appropriate scalar or 2-D features from data, handling, therefore, different and multiple types of information for classification purposes, while providing efficient solutions due to their massive and distri-

buted parallelism [7]. In particular, feedforward multilayer perceptrons, usually trained by some backpropagation variant, as well as probabilistic neural networks have been proposed as powerful tools for image recognition [17] and texture segmentation problems [1, 9, 3]. Multilayer perceptrons are capable of approximating arbitrary nonlinear mappings. However, there are a number of practical concerns when using these networks: a first has to do with the choice of the appropriate network size which is unknown in most applications; a second is the required training time which can be very long; another has to do with network generalization i.e. with the performance of the network when presented with data outside its training set, which can be very poor.

Various techniques have recently been proposed for overcoming those problems, including pruning or constructive techniques during training and derivation as efficient backpropagation variants. These aspects, how-

ever, still constitute open research problems in the design of appropriate multilayer feedforward networks. On the other hand probabilistic neural networks can approach optimal minimum risk decision surfaces [18] and seem to overcome the aforementioned disadvantages but suffer from huge memory requirements. Recent studies concerning the problem of image classification indicate that structured neural network classifiers are especially useful when applied directly to image pixel values, and not to a set of features extracted from the images [12, 3]. Multilayer probabilistic perceptrons have been recently examined in the neural network field, as a tool for image classification [1, 5]. In this paper neural networks are used to correlate the multichannel information from range, intensity and possibly color images, while multiresolution analysis is used to reduce the dimensionality of the acquired representations.

The overall structure of this paper is as follows . Section 2 presents a brief description of the multiresolution analysis that is used in the paper. Section 3 presents a brief description of probabilistic neural networks, while section 4 presents the proposed architectures for multiresolution analysis of multichannel images. Simulation studies are presented in section 5 , and conclusions are given in section 6 of the paper.

2 2-D Multiresolution Analysis

Multiresolution image representations have been known for a long time in applications such as correlation matching, edge detection, segmentation and image analysis [4, 20]. These representations are generally used to reduce the problem dimensionality and the associated computational load and to perform feature extraction at different resolution levels. Resolution reduction is most commonly performed by subsampling or local averaging of the image pixel values.

Multiresolution image analysis and processing based on the use of the wavelet/subband decomposition has recently attracted major interest, mainly for coding and scalability applications [14, 2]. In general, the transition from one level of resolution to a lower one is implemented by subsampling each dimension of the higher resolution data, usually by a factor of two, and by using a finite impulse response (FIR) filter with fixed taps; typical 8 or 32 tap filters are given in [2] . As a consequence, 4 ($N/2 \times N/2$) low resolution images are produced from each color component of a 2-D ($N \times N$) color high resolution image. One of these four images corresponds to the low frequency content of the high resolution 2-D information. By keeping only this component, the procedure can be repeated, until a res-

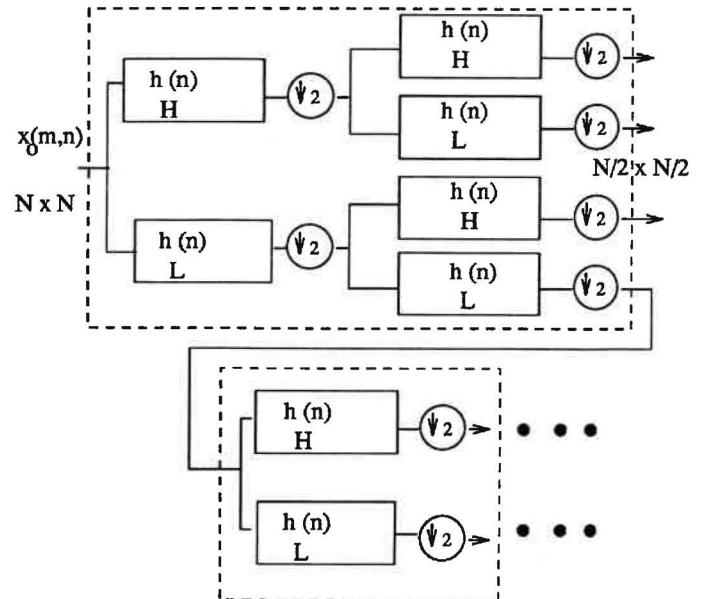


Figure 1: Multiresolution Image Analysis

olution level, usually defined by experimental study, is reached, below which significant features cannot be extracted and are disregarded. This process is depicted in Fig. 1. Let \mathbf{x}_0 denote an $N \times N$ 2-D image. Then using finite impulse response (FIR) filters $h_L(n)$ and $h_H(n)$, where $h_L(n)$ is a low-pass and $h_H(n)$ is a high-pass filter, we can split the original image into four lower resolution $\frac{N}{2} \times \frac{N}{2}$ images. Applying, for example, the low-pass filter $h_L(n)$ in the horizontal and then in the vertical direction of the original image (we consider the separable case for simplicity), we get the *approximation* image at the lower resolution level $j = -1$, denoted as

$$\mathbf{x}_{-1}^{LL}(m, n) = \sum_{k=1}^N \sum_{l=1}^N h_L(2m - k)h_L(2n - l)x_0(k, l) \quad (1)$$

By applying all other possible combinations of the above FIR filters $h_L(n)$ and $h_H(n)$, we get three lower resolution *detail* images, denoted as $\mathbf{x}_{-1}^{LH}, \mathbf{x}_{-1}^{HL}, \mathbf{x}_{-1}^{HH}$. Moreover, if the above procedure is successively applied to the *approximation* images we have a *multiresolution approximation* of the original image, providing images of continuously decreasing size.

Each decimation procedure from a data representation of higher to a corresponding one of lower resolution introduces a loss of information. It is desired that this loss be as small as possible. An adaptive scheme which forms the hierarchical representation by selecting at each level the image with the greater information content instead of the one with the low frequency content based on the minimization of the error between

the original and some low resolution level is presented in [21]. The filters designed with this method provide highly decorrelated images appropriate for image classification at lower resolution.

3 Probabilistic Networks

Many neural network classifiers provide outputs which estimate Bayesian *a posteriori* probabilities. When the estimation is accurate, network output values sum to one and can be treated as probabilities. Bayesian probabilities can be estimated by multilayer perceptrons where the desired network outputs correspond to, say, M classes, one output is equal to unity, and all other are equal to zero, while a mean squared error or a cross entropy cost function is minimized by the network[11]. Error feedback supervised algorithms using the Kullback Leibler (KL) criterion, with generalized sigmoidal [13, 11], have been shown capable of producing Bayesian *a posteriori* probabilities or conditional likelihood estimates for classification purposes. The estimation accuracy generally depends on the network complexity, the amount of training data, and the degree to which training data reflect true likelihood distributions and *a priori* class probabilities.

Unlike perceptron type networks, which classify input vectors by learning multidimensional decision surfaces, probabilistic neural networks (PNNs) [18] classify input vectors by forming non parametric probability density functions (p.d.f.). The network structures are similar to that of multilayer perceptrons; the primary difference is that the sigmoid activation function is replaced by the exponential one. Key advantages of PNNs are that training requires only a single pass and that decision surfaces approach the Bayes-optimal decision boundaries as the number of training samples grows.

PNNs utilizes the fact, that in the limit, any smooth and continuous p.d.f, say f_A of a class of multidimensional data \mathbf{X} can be estimated by a sum of multivariate Gaussian distributions centered at each training sample [16]; $f_A(\mathbf{X})$ will be equal to :

$$\frac{1}{(2\pi)^{p/2}\sigma^p} \frac{1}{m} \sum_{i=1}^m \exp[-(\mathbf{X} - \mathbf{X}_{\alpha i})^t (\mathbf{X} - \mathbf{X}_{\alpha i})/2\sigma^2] \quad (2)$$

where

i = pattern number,

m = total number of training patterns,

$\mathbf{X}_{\alpha i}$ = i th training pattern from category A

p = dimensionality of input vector \mathbf{X}

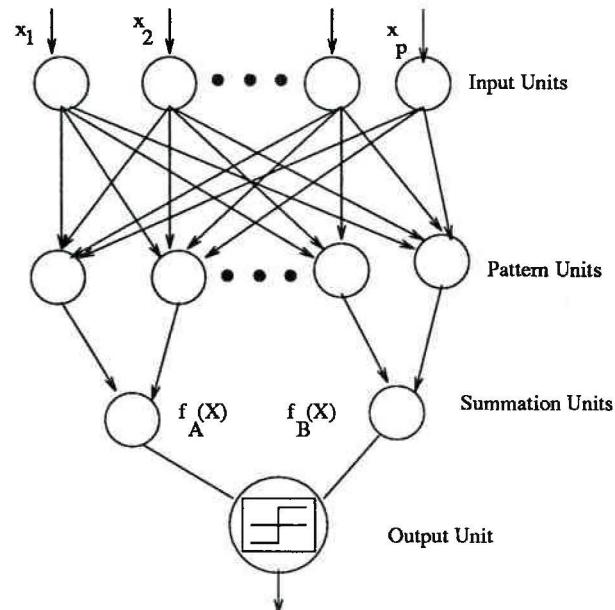


Figure 2: Probabilistic Neural Network for classification of patterns into two categories

σ = "smoothing parameter".

Fig. 2 shows a neural network architecture which can be used for estimation of the p.d.f of classes of input patterns \mathbf{X} and subsequent classification of them into categories. A two category classification into classes A and B is shown in Fig. 2; extension to a larger number of classes is straightforward. The input units are merely distribution units that supply the p elements of each input data vector to all pattern units, the number of which is equal to the number m of training data. The i th pattern unit forms a dot product of the input data vector and a corresponding weight vector \mathbf{W}_i and passes it through the exponential activation function, as follows:

$$\exp[(\mathbf{W}_i \mathbf{X}^t - 1)/\sigma^2] \quad (3)$$

Assuming that both \mathbf{X} and \mathbf{W}_i are normalized to unit length, the output of the i th pattern unit is equivalent to a probability measure:

$$p_i(\mathbf{X}) = \exp[-(\mathbf{X} - \mathbf{W}_i)^t (\mathbf{X} - \mathbf{W}_i)/2\sigma^2] \quad (4)$$

The network summation units compute through Eq. (4) each class p.d.f as represented by Eq. (2). The output units simply select the class with the maximum p.d.f.

The network is trained by setting the \mathbf{W}_i weight vector of the i th pattern unit equal to the corresponding

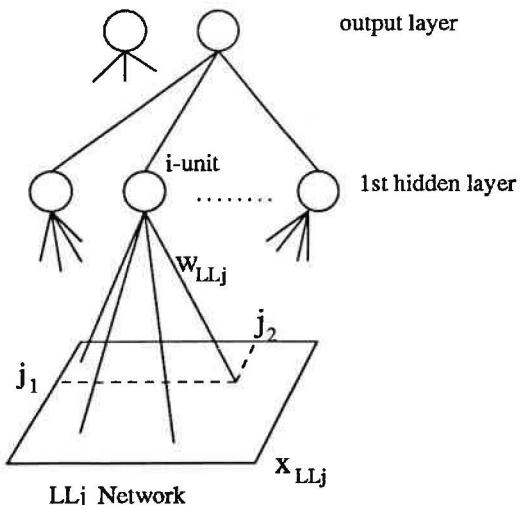


Figure 3: The neural network at level j , for classifying the *approximation* images of that level.

\mathbf{X} pattern in the training set, and then connecting the pattern unit's output to the appropriate summation unit. The smoothing parameter σ controls the activation function; large (small) values of σ reduce (increase) the sensitivity of the exponential function.

Both multilayer perceptrons and probabilistic networks are considered in the multiresolution network framework introduced in the next section for efficient classification of multichannel images.

4 Neural Networks for Multiresolution Classification

In the following we develop two different neural network architectures which take advantage of the multiresolution analysis described in section 2. Let us assume that a feedforward neural network is used to classify the original \mathbf{x}_0 $N \times N$ image. Then, network training can be performed at multiple resolution levels, starting from some low resolution level which sufficiently describes the problem. More specifically, the proposed procedure starts by training a network at, say, resolution level j with $j \leq -1$, (network LL_j) to classify *approximation* images \mathbf{x}_j^{LL} at that resolution level (see Fig. 3). After training, the network performance is tested, using a test set of *approximation* images at the same resolution level j . If the performance is not acceptable, training is repeated at the next higher resolution level, i.e. at level $j + 1$.

At this point we deviate the development of the multiresolution neural networks architectures for multilay-

er perceptrons and for single layer probabilistic neural networks.

4.1 Hierarchical Multilayer networks

Let us assume that after network training at a particular resolution level, its generalization is examined and found not satisfactory, using, for example, a validation data set. In this case, the method includes a projection of all weights, or at least of the weights of connections between the input and first hidden layer (which usually constitute the majority of network weights) towards the next resolution level, so that the already derived network knowledge, i.e. weight values, be included in the network architecture of the following level (a similar approach is described in [6]). This approach constructs an hierarchical network, reducing the required training times as well as the number of free parameters, i.e. weights, that are to be determined at each resolution level.

Since the information of an (*approximation*) image at level $j + 1$ is equivalent to the information included at both the *approximation* and *detail* images at level j , we can also train three more networks (LH_j, HL_j, HH_j), separately (or in parallel) from the LL_j one, (as was presented in Fig. 3), to classify the *detail* images at level j . If it is required to repeat the training procedure at level $j + 1$ it will be desired that the network at level $j + 1$ (network LL_{j+1}) a-priori includes as much as possible from the "knowledge" of the problem acquired by the former networks at level j (networks LL_j, LH_j, HL_j, HH_j). The training procedure that we propose uses first a rather simple fully connected feedforward neural network, to classify a quite low resolution representation of the original image and then recursively constructs the network architecture so as to handle the image at higher resolution levels. This can be accomplished using a network, at level LL_{j+1} , the first hidden layer of which consists of the union of the sets of hidden units of the lower resolution networks (see Fig. 4). Moreover, we impose the constraint that the inputs to the units of the first hidden layer of network LL_{j+1} be identical to the inputs of the corresponding units of the networks at level j . This constraint is used next to express that part of network LL_{j+1} corresponding to network LL_j . Let the input to the i -th unit of the first hidden layer of network LL_j be

$$\sum_{j_1=1}^{N/2} \sum_{j_2=1}^{N/2} w_j^{LL}(j_1, j_2) x_j^{LL}(j_1, j_2) \quad (5)$$

Then for the LL_{j+1} network, the input to the i -th of the set of units that correspond to the LL_j network

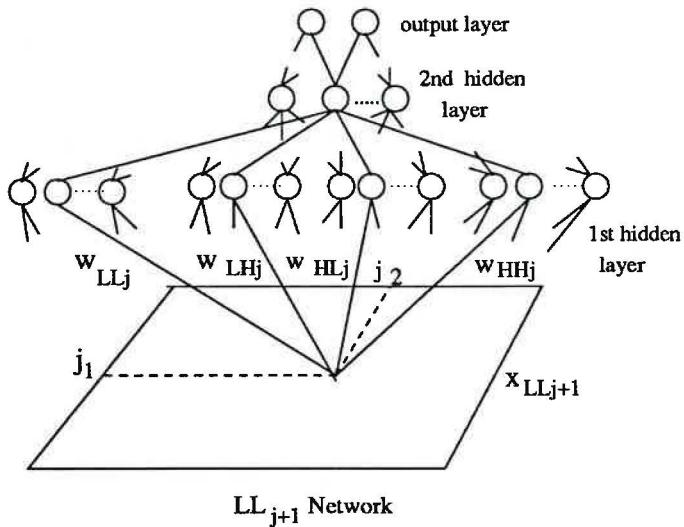


Figure 4: Neural network at level $j + 1$.

will be analogously

$$\sum_{k_1=1}^N \sum_{k_2=1}^N w_{j+1}^{LL}(k_1, k_2) x_{j+1}^{LL}(k_1, k_2) \quad (6)$$

If the computed values in (5) and (6) are required to be equal to each other, then it can be easily shown using (1) that

$$w_{j+1}^{LL}(k_1, k_2) = \sum_{j_1=1}^{N/2} \sum_{j_2=1}^{N/2} h_L(2j_1 - k_1) h_L(2j_2 - k_2) w_j^{LL}(j_1, j_2) \quad (7)$$

A similar form can be derived relating weights in networks LH_j, HL_j, HH_j and the corresponding sets of units of the network LL_{j+1} .

The above forms permit computation of the generally large number of weights between the network input and first hidden layer be efficiently performed at lower resolution; computation of the generally less complex upper hidden layers, which are required for extracting both the coarse and fine input information, will then be performed by training the corresponding part of higher level network LL_{j+1} .

When training the network at each resolution level we adopt the idea of cascade correlation regarding the network size ; we start with a small network size and gradually increase it by adding new units to it during its training. Regarding training time, efficient variants of the backpropagation algorithm [10] can be used to obtain generally fast and smooth convergence of the learning process. Finally, regarding network generalization, it is well known that it heavily depends on the use of a rather small, compared to the number of

training samples, number of weights (as recent results based on the Vapnik-Chervonenkis dimension indicate [15]). In our approach, by reducing the size of 2-D input images using the above-described multiresolution technique, we also reduce the size of the network that receives these representations. Moreover, network training is performed, starting from the lower resolution level and gradually moving towards the fine detail representations.

4.2 Hierarchical probabilistic networks

The introduction of multiresolution analysis in multi-layer perceptrons results in reduced training time because of the association of the weights between the different resolution levels. Probabilistic neural networks have, on the other hand, serious advantages with respect to training time, since they are single layer networks with a single pass training procedure. Their main drawback is the large memory requirements, since during training all the training samples are stored in the weights of pattern units. If the training samples are 2-D gray scale, color, or 3-D range data, then the memory requirements can grow enormously.

Consequently, it is desirable to reduce the memory requirements with as much as possible less degradation of the network performance. Multiresolution analysis can be used to reduce the dimensionality of the input images, by choosing a subsampled and filtered replica of the original image (thus reducing the number of the weights of pattern units); moreover this replica could be "forced" to contain the maximum information quotient using appropriate filterbanks [21]. In this way it is possible to optimally select the "knowledge" from the low resolution networks with as least as possible loss of information in the low resolution representations.

Let us assume that the image \mathbf{X} at some resolution level $j + 1$ is splitted in two parts $\mathbf{X} = [\mathbf{X}_a \mathbf{X}_b]$, the first of which represents the *approximation image* \mathbf{x}_j^{LL} , and the second the remaining *detail images* $\mathbf{x}_j^{LH}, \mathbf{x}_j^{HL}, \mathbf{x}_j^{HH}$ subsampled replicas of the original image. Let training be performed at resolution level j using the *approximation image*, i.e., \mathbf{x}_j^{LL} . As was described above, if the network performance is not acceptable, the *detail images* can be classified by a second network that is separately trained; the outputs of the two networks are combined to yield classification at level $j + 1$.

Following the decomposition of the input vector \mathbf{X} ; a corresponding decomposition of the network weights $\mathbf{W}_i = [\mathbf{W}_{ai} \mathbf{W}_{bi}]$ can be performed. The output of the i th pattern unit $p_i(\mathbf{X})$ can be consequently decomposed in the following form:

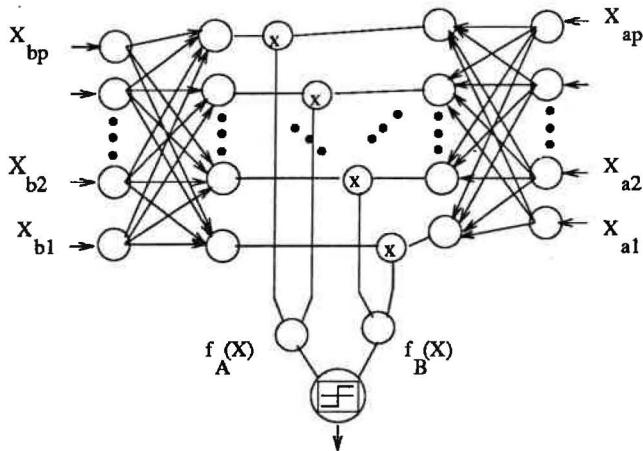


Figure 5: Hierarchical probabilistic network classifier.

$$\begin{aligned}
 p_i(\mathbf{X}) &= \exp[(\mathbf{W}_i \mathbf{X}^t - 1)/\sigma^2] \\
 &= \exp[(\mathbf{W}_{ai} \mathbf{W}_{bi})[\mathbf{X}_{ai} \mathbf{X}_{bi}]^t - 1)/\sigma^2] \\
 &= k \cdot \exp[(\mathbf{W}_{ai} \mathbf{X}_{ai} - 1) + (\mathbf{W}_{bi} \mathbf{X}_{bi} - 1)]/\sigma^2 \\
 &= k \cdot p_{ai}(\mathbf{X}) \cdot p_{bi}(\mathbf{X})
 \end{aligned} \quad (8)$$

where k is a scaling factor equal to $\exp(1/\sigma^2)$.

Thus in order to combine the probabilistic networks trained to classify the *approximation* and *detail* images at level j , we simply have to multiply the outputs of their respective pattern units and then feed them in the appropriate summation units, as depicted in Fig. 5.

5 Simulation results

Simulation results are presented, using real multichannel image data, which illustrate the performance of the various techniques proposed in this paper. It is shown that the use of neural networks in the described multiresolution formulation can provide an effective and reliable procedure for multichannel image classification and recognition.

A real life application was chosen to examine the performance of the proposed multiresolution classification procedure, namely the inspection of solder joints in printed circuit board manufacturing. 2-D binary scale images (23×23 pixels), showing the height and the intensity as functions of the position across solder joints, were obtained by an optical laser scanner and used as a multichannel signal to be classified in two categories; namely, good or poor solder joints, the latter containing insufficient amount of solder. An example of a good and of a defective solder joint (height

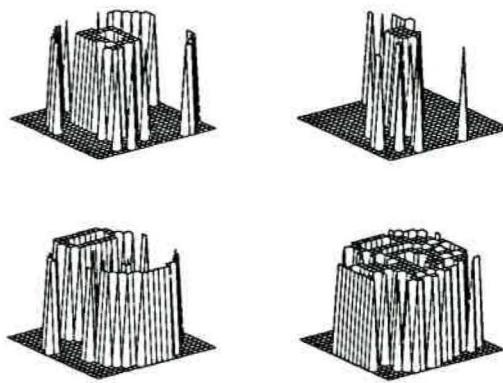


Figure 6: (a) : a good solder joint (intensity/height), (b): a defective one

and amplitude representation) is shown in Figs. 6a , 6b respectively. A database of 50 good and 50 poor solder joint images was used in our experiments. 25 good and 25 poor solder joints (both height and amplitude) were selected to form the learning set and the rest for the test set.

First we used a two-hidden-layer fully connected feedforward neural network with an input layer of $2 * [(23)^2] = 1058$ units, and trained it using a learning set of the original 23×23 intensity and height pixel images. The large number of interconnection weights, in comparison with the small number of training patterns, resulted in unsuccessful learning. The network, after long training time, was unable to classify correctly the training set; a classification rate of only 72 % was obtained.

Then, we used the FIR 8-tap QMF filters $h_L(n) = [0.0094, -0.0707, 0.0694, 49, 0.49, 0.0694, -0.0707, 0.0094]$, and $h_H(n) = (-1)^{1-n} h_L(1-n)$ combined with a subsampling by 2 to get images at each lower resolution level. The details of the already small in size original images are sufficiently preserved only at resolution level $j = -1$. The resulting images corresponding to Figs. 6a,b are shown in Figs. 7a,b. For this reason we tried to perform the classification at this level.

We used four networks of the form of Fig. 3, to classify independently the LL_{-1} , LH_{-1} , HL_{-1} and HH_{-1} images at this resolution level. Each network had 121 units at the input layer, since the dimension of the images at level $j = -1$ was 11×11 pixels. The LL_{-1} network was the most successful one, classifying the corresponding test set with 88% accuracy. This difference can be explained by the fact that the *approximation* image contains more information than the *detail*

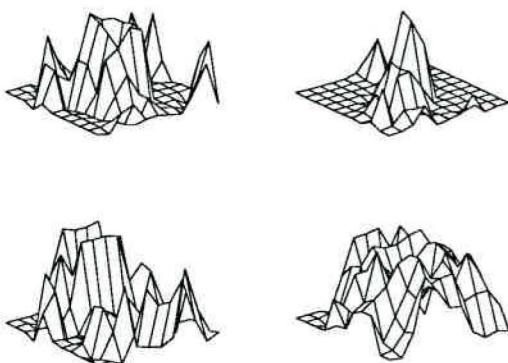


Figure 7: Corresponding representations at lower resolution

one at this level. Then, we formed the network of the next resolution level $j = 0$ (as shown in Fig. 4), by combining the two previous networks. The weights between the input and the first hidden layer obtained as in Eq. (7), remained fixed. The network was trained with the original training set and classified the original test set with accuracy 94%.

In the following we performed a similar experiment using probabilistic neural networks. First we tested the performance of PNNs using the original images. A classification rate of 79 % was achieved using as many as 1058 stored weights in the pattern units. The experiment was repeated next separately on the *approximation images* and the *detail images* achieving classification rates of 86.6 % and 66.7 % using 242 and 726 stored weights results respectively. Using a two resolution level probabilistic network as described in the previous section, a 94 % classification rate was also achieved. Different smoothing parameters were used for the two probabilistic neural networks, being $\sigma_1 = 0.0001$ and $\sigma_2 = 0.01$ respectively. A smaller smoothing parameter was used in the first network than in the second one, based on the fact that the classification rate obtained by the former network was larger than the rate obtained by the second one when operating independently.

6 Conclusions

Multiresolution feedforward neural network architectures were introduced in this paper as a means for efficient classification of multichannel images. Both multilayer backpropagation networks and probabilistic networks were considered. Multiresolution techniques

were developed, which were shown to increase the performance of both networks. In the case of multilayer perceptrons reduced training times were achieved and a method for transferring "knowledge" across different resolution levels was developed. In the case of probabilistic neural networks a considerable reduction of the memory requirements was achieved, alleviating one of the most crucial problems of these networks.

We are currently investigating the use of the proposed networks for fast retrieval of stored images from large image databases organized according to multiresolution image representations.

References

- [1] M. Morrison and Y. Attikiouzel, "A Probabilistic Neural Network Based Image Segmentation Network for Magnetic Resonance Images", *IJCNN 92*, Portland, 1992.
- [2] I. Daubechies et al, "Image coding using the wavelet transform", *IEEE Trans. on Image Processing*, vol. 1, pp. 205-220, 1992.
- [3] A. Delopoulos, A. Tirakis and S. Kollias, "Invariant Image Classification using triple-correlation Neural Networks", *IEEE Trans. on Neural Networks*, May 1994.
- [4] R. Fisher, "From surfaces to objects, computer vision and three dimensional analysis", *Wiley*, 1989.
- [5] J. Hwang, E.Y. Chen and A.F. Lippman, "Probabilistic Image Modelling via neural Networks" *ICASSP 93* Minneapolis, Minnesota.
- [6] C. Hand et al, "A neural-network feature detector using a multiresolution pyramid", in R. Linngard, ed., *Neural networks for vision, speech and natural language*, Chapman and Hall, 1992.
- [7] D. R. Hush and B.G. Horne, "Progres in Supervised Neural Networks", *IEEE Signal Processing Magazine*, pp. 8-39, Jan 1993.
- [8] I. Ng, J. Kittler and J. Illingworth, "Supervised segmentation using a multiresolution data representation", *Signal Processing*, vol. 31, pp. 133-164, 1993.
- [9] S. Kollias ans D. Kalogeras, " A Multiresolution Probabilistic Neural Network for Image Segmentation", *ICASSP-94*, Adelaid, 1994.

- [10] S. Kollias and D. Anastassiou, "An adaptive least squares algorithm for the efficient training of multilayered networks", *IEEE Trans. on CAS*, vol. 36, pp. 1092-1101, 1989.
- [11] R. P. Lippmann, "Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities", *Neural Computation*, vol. 3 pp. 461 - 483, 1992.
- [12] Y. LeCun, "Generalization and Network Design Strategies", *Connectionism in Perspective*, North Holland, Switzerland, pp. 143 - 155. 1989.
- [13] J. Makhoul, "Pattern Recognition Properties of Neural Networks", *IEEE-SP Workshop on Neural Networks for Signal Processing*, pp. 173-186, Princeton NJ, 1991.
- [14] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE Trans. on PAMI*, vol.11, pp. 674-693, 1989.
- [15] J. Moody and C. Darken, "Fast learning in networks of locally-tuned processing elements", *Neural Computation*, vol. 1, pp. 281-293, 1989.
- [16] E. Parzen "On estimation of a probability density function and mode", *Ann. math Stat.*, vol. 33, pp. 1063-1076, 1965.
- [17] Y. Pao, "Adaptive pattern recognition and neural networks", *Addison Wesley Co.*, 1989.
- [18] D. F. Specht, "Probabilistic Neural Networks and the polynomial Adaline as Complementary Techniques for Classification", *IEEE Trans. On Neural Networks*, vol. 1, no. 1, Mar. 1990.
- [19] P. Woods et al, "The use of geometric and gray level models for industrial inspection", *Pattern Recognition Letters*, pp. 1-17, 1987.
- [20] S. Tanimoto and T. Pavlidis, "A hierarchical data structure for picture processing", *CGIP 4*, pp. 104-109, 1975.
- [21] A. Tirakis A. Delopoulos and S. Kollias, "2-D Filter Bank Design For Optimal Subband Reconstruction", *IEEE Trans on Image Processing*, to be published.

A Back-Propagation Adaptation Yielding a Ternary Output ANN With Limited Integer Weights

G.R. Cukowicz & M.R. Haskard

MicroElectronics Centre

University of South Australia

The Levels, South Australia

email: grc@awam.com.au, m.haskard@unisa.edu.au

Abstract

This paper presents an adaptation of the back-propagation algorithm which results in a network that is described by a set of finite integer weight values and ternary activation function. Training requires full floating point processing to be performed off line. The resulting trained network can be implemented as a digital VLSI circuit using simple logic and memory structures. The novelty of the research lies in the use of a ternary activation function as opposed to binary, yielding greater network versatility with minimal expense in hardware overhead. Discussion concentrates on presenting the algorithm required for training, the conditions under which the algorithm is successful and the primary advantage of the ternary activation function.

1. Introduction

Artificial neural networks (ANN's) have found wide application in areas which have challenged conventional computing. As a result of this success, much research effort has been expended on the development of new paradigms with primary regard given to the resulting computational abilities. Unfortunately, as noted by Shoemaker et. al. [6], the majority of this research has neglected the problems associated with hardware implementation of the resulting networks as a parallel system.

Whilst silicon is not likely to solve all the problems associated with hardware implementation of ANN's, it appears to be the best solution available in the near future [4]. As a result, it is believed that more regard should be given to the development of algorithms which result in networks that are realisable as VLSI parallel processor arrays. This paper deals with the adaptation of the back-propagation algorithm presented by McLelland et. al. [10], one of the most commonly used and extensively studied ANN paradigms. The resulting digital network is represented by a limited set of integer weight values and the outputs of the processing elements are restricted to three values; -1, 0 and +1 (no, uncertain and yes). It should be noted that the training procedure requires full floating point arithmetic and it is only the resulting network that is implemented in hardware.

The primary distinction between the research presented in

this paper and that presented in previous papers [6,7,8] is in its use of a ternary activation function as opposed to a binary function. The results demonstrate that, for a particular class of output codings referred to as 1 of x type representations, the ternary activation scheme has the potential to improve the generalisation yield of resulting networks.

2. The Proposed Adaptation

The back-propagation algorithm is based on a mathematical theory of gradient descent [10] in overall output error. This theory requires a differentiable non-decreasing activation function, the most commonly used being the Sigmoid. In addition, the weight changes are based on the derivative of this function resulting in a continuous range of weights. It is the storage and calculation of these continuous values that pose the problem for hardware implementation.

The typical software solution overcomes this problem by using a floating point processor and performing all calculations sequentially. However, as a result of the silicon area occupied by floating point processors, this is not a feasible option for large scale parallel processing. Various approaches to overcome this problem have been reported [1,2,3,6,7,8,9,11] ranging from analog to digital to a mixture of the two. Whilst analog circuits are simple and implementations of analog multipliers/accumulators are fast [2], they are not very densely integrated and are prone to noise, inter-chip and inter-wafer variations [3]. As a result, the approach taken has been to implement a fully digital ANN.

As previously mentioned, the proposed system uses a limited set of integer weight values and a ternary processing element output. Figure 1 shows a block diagram of the proposed processing element consisting of a static RAM for storage of the integer weights, a simple combinational logic multiplier, accumulator and threshold circuit. External to the processing element are weight and processing element decoders plus additional control circuitry for read/write operations. The advantages of the

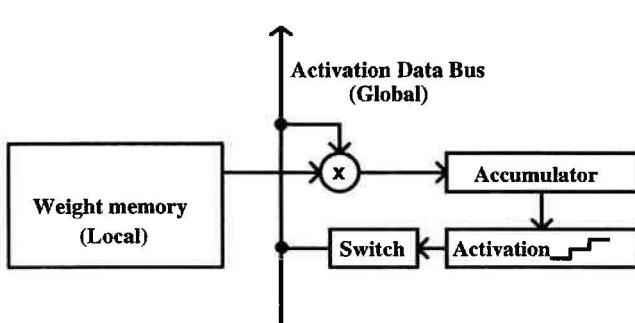


Figure 1. Processing element block diagram.

proposed system are substantially less silicon area required per processing element due to the simple number representation system and threshold circuitry used as the activation function.

3. Training Algorithm

The training algorithm is heavily based on work presented by Marchesi et. al. [8] and requires floating point processing. The algorithm consists of two phases. The primary phase consists of the standard back-propagation algorithm using the Sigmoid activation function and results in floating point weights. A deviation where the single Sigmoid was replaced by two Sigmoids more closely approximating the ternary activation function was also investigated, the reasoning being that a closer approximation would lead to a better retention of properties after quantization. Unfortunately results showed that, whilst there was little difference in secondary phase trainability for networks originating from the two different activation functions, the generalisation capabilities of the quantized networks resulting from the dual Sigmoid approach were considerably worse than those using the conventional Sigmoid activation function.

The secondary phase takes the network generated by the primary phase and continues training whilst progressively decreasing the number of allowable quantized weight values. The training procedure uses normalised weight values which are scaled back up to integer values at the completion of training. Prior to a description of the secondary training algorithm, several key definitions need to be made.

- The number of bits used for weight representation at a given point in time is denoted n .
- The final set of allowable quantised integer weight values, W_l , is defined by equation (1).

$$W_l = \{x : -2^{n-1} \leq x \leq 2^{n-1}\}, x \in I \quad (1)$$

- The ternary activation (threshold) function to be implemented in hardware is defined by equation (2).

$$a_t(\text{sum}) = \begin{cases} -1 & \text{if sum} \leq -(t+1) \\ +1 & \text{if sum} \geq t \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The region of sum values which results in an activation value of 0 is referred to as the uncertainty region whilst the value t is referred to as the positive threshold point. It should be noted that t is always a power of 2 and the negative threshold point occurs at a value of $-(t+1)$ due to hardware implementation considerations.

- During training, the number of bits allowable for weight representation is reduced in powers of two. The initial number of bits to be used is denoted n_i , the final number of bits is denoted n_f .

The secondary phase training algorithm is as follows.

1. Retrieve the network information generated by the primary training phase.

2. Normalise the weights incident on each processing element (PE). This requires the determination of the maximum weight magnitude, W_{Hj} , incident on processing element j where W_{Hj} is described by equation (3), w_{ji} is the weight value associated with the path from processing element i in the previous layer to processing element j in the current layer, and K is the number of

$$W_{Hj} = \max\{|w_{ji}|, i = 1, 2, \dots, K\} \quad (3)$$

processing elements in the previous layer. The weights are then normalised by dividing each weight incident on PE j by W_{Hj} , the new weights being denoted w'_{ji} .

3. Set the initial number of bits to be used for weight representation to $n = n_i$.

4. Scale the weights in such a way that the retention of relationships between weights after quantisation is optimised whilst the resulting error in output sum is minimised. Equation (4) describes the error introduced by

$$E_j(A) = \sum_{i=1}^K (w'_{ji} - \langle A \cdot w'_{ji} \rangle)^2 \quad (4)$$

weight scaling and quantisation where $\langle \cdot \rangle$ denotes rounding to the nearest allowed weight value in W_N and A is the scaling factor. W_N is defined by equation (5).

$$W_N = \frac{W_l}{2^n} \quad (5)$$

It should be noted that the activation function used during this training stage is also scaled by dividing both positive and negative threshold points by 2^n . Note that it is the final number of bits for weight representation that determines the dividing factor and therefore must be decided on prior to commencement of the secondary training phase.

Determination of the optimum value of A , A_j , for a given PE j is performed empirically by calculating $E_j(A)$ for values of A in the range [0.5, 1.5] incrementing in steps of 0.001. Once A_j has been determined, scaling is performed according to equation (6) where w''_{ji} represents the scaled quantized weight value.

$$w''_{ji} = \langle A_j \cdot w'_{ji} \rangle \quad (6)$$

5. Present the training set to the new network. If the resulting error is non-zero (remembering that the activation values are now exact) then further training steps are required. Further training is achieved by calculating an accumulated weight change value at the end of a complete training set presentation. The accumulated weight change value is described by equation (7) where p indicates the p 'th input/output pair of the training set, E is the number of input/output pairs in the training set, and δ_j is described by

$$\frac{\Delta w_{ji}}{E} = \left[\sum_{p=1}^E (\delta_j \cdot a_i) \right] \quad (7)$$

equation (8) if processing element j is in the output layer or equation (9) if processing element j is a hidden element.

$$\delta_j = (t_j - a_j) \quad (8)$$

$$\delta_j = \left[\sum_k \delta_k \cdot w_{kj} \right] \quad (9)$$

At the end of a complete training set presentation, weight changes are made according to equation (10) where η is a

$$w''_{ji}(m+1) = \left\langle w''_{ji}(m) + \eta \cdot \Delta w_{ji} \right\rangle \quad (10)$$

learning rate coefficient. In the event that the RMS error does not change (which indicates that weight changes are not having a significant effect), η is increased until the RMS error does change. Training set presentations are continued until the RMS error reaches zero. The initial value of η was 0.2 and the increment 0.02.

6. If $n \neq n_f$ then n is decremented by one and the procedure continues from step 4. Otherwise training is complete in which case the normalised weight values are multiplied by 2^n to produce integer values.

This completes the description of the training procedure.

4. Experiments

In order to evaluate the training procedure, experiments were carried out on a relatively simple digit recognition problem. There were several objectives to the evaluation. Firstly, an investigation into the effects of the number of bits used for weight representation and the width of the uncertainty region on the trainability and generalisation capabilities of the resulting networks was required. The purpose was to gain some insight into what combination of these two factors should be implemented in hardware. In addition to this, the effectiveness of the ternary activation function as a means of increasing generalisation yield for noisy input patterns on 1 of x type output codes was investigated, this being considered one of the prime advantages of the ternary activation system.

The data sets for training and evaluation are shown in figure 2. Rather than using a binary input representation,

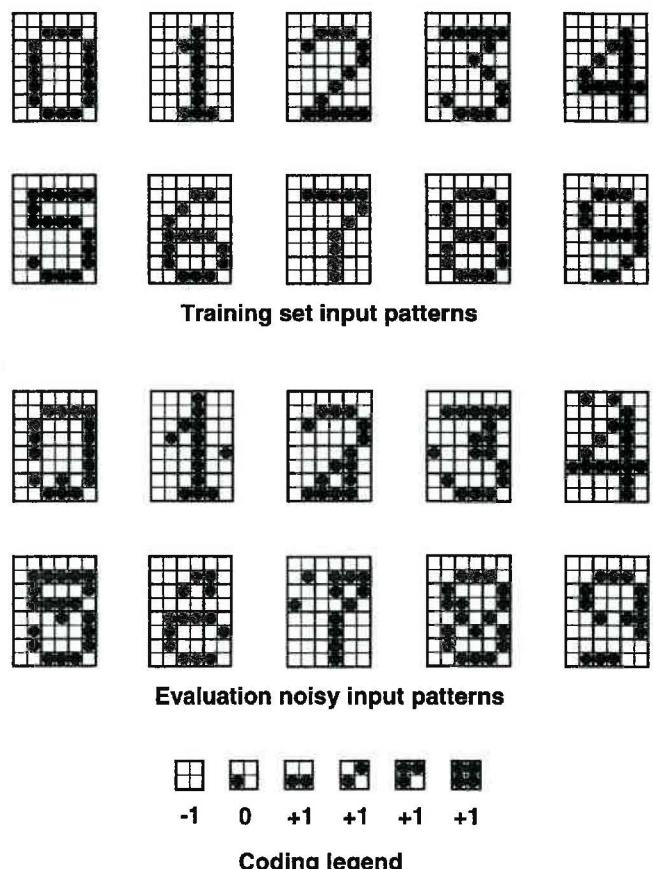


Figure 2. Training and evaluation data sets.

the ternary nature was taken advantage of to reduce the number of elements in the input vector from 48 to 12, thus reducing the number of processing elements required in the input layer. This was achieved by dividing the 48 element array into 12 sub-arrays of 4 elements each with coding as shown in Figure 2.

In order to gain more insight into the effectiveness of the training algorithm and properties of the resulting networks, three different output representation methods were used. The output representations are shown in Table I. Training was performed for hidden layer sizes ranging from 5 to 20 with 5 different random starting points for each hidden layer size in order to ensure that the observed results were a good representation of the overall performance of the system. The number of bits considered for weight storage ranged from 4 to 10 whilst the positive threshold points were varied from 1 to 32 in powers of two.

Digit	Binary code	Ternary code	1 of 10 code
0	-1 -1 -1 -1	-1 -1 -1	-1 -1 -1 -1 -1 -1 -1 -1 -1 +1
1	-1 -1 -1 +1	-1 -1 0	-1 -1 -1 -1 -1 -1 -1 -1 -1 +1 -1
2	-1 -1 +1 -1	-1 -1 +1	-1 -1 -1 -1 -1 -1 -1 +1 -1 -1 -1
3	-1 -1 +1 +1	-1 0 -1	-1 -1 -1 -1 -1 -1 +1 -1 -1 -1 -1
4	-1 +1 -1 -1	-1 0 0	-1 -1 -1 -1 -1 +1 -1 -1 -1 -1 -1
5	-1 +1 -1 +1	-1 0 +1	-1 -1 -1 -1 +1 -1 -1 -1 -1 -1 -1
6	-1 +1 +1 -1	-1 +1 -1	-1 -1 -1 +1 -1 -1 -1 -1 -1 -1 -1
7	-1 +1 +1 +1	-1 +1 0	-1 -1 +1 -1 -1 -1 -1 -1 -1 -1 -1
8	+1 -1 -1 -1	-1 +1 +1	-1 +1 -1 -1 -1 -1 -1 -1 -1 -1 -1
9	+1 -1 -1 +1	+1 -1 -1	+1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

Table I. Output representation methods.

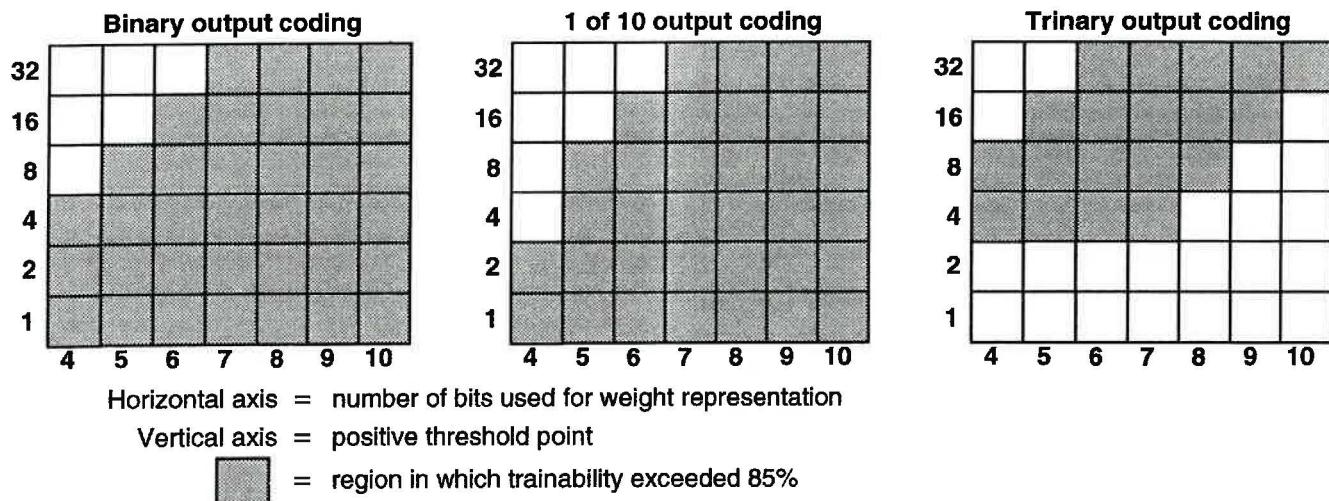


Figure 3. Trainability as a function of weight size.

4.1. Trainability

The results of the trainability investigations are shown in Figure 3, with shaded regions showing the combinations of weight size and uncertainty region width for which more than 70 out of 80 networks successfully completed the secondary training phase. The results for the binary coding and 1 of 10 output representations were found to be quite similar; not surprising since both use only binary values in their output vectors (± 1). For this type of output representation, trainability was found to decrease for ratios of maximum weight value to uncertainty region widths above a certain level. In this particular case, that ratio is approximately 1:1. For a ratio of greater than 1:1, each weight has the ability to cause the sum to exceed the threshold value thus resulting in the required activation value of ± 1 . As the width of the uncertainty region is increased, the ability of the weights to result in sums which lie outside the uncertainty region is decreased.

Turning our attention to the ternary output coding, a second region of poor trainability is observed for ratios of maximum weight magnitude to uncertainty region width

below a certain value. For this particular problem, the ratio is approximately 6:1. This region results from a lack of ability to result in sums which fall in the uncertainty region, with changes in weights likely to cause sum values which oscillate about the uncertainty region but never fall into it.

4.2. Generalisation Capabilities

The noisy input patterns used for investigating the generalisation capabilities of the resulting networks are shown in Figure 2 with the results being shown in Figure 4. Each pattern contains three randomly chosen altered elements. The criteria used to define good generalisation was not an absolute value, but rather a combination of a relatively stable percentage of correct mappings and less than 5% of examples being incorrectly mapped to other known output codes. The charts showing the dependence of generalisation capabilities on the number of bits used for weight representation and the width of the uncertainty region show similar properties to those of the trainability charts with one significant difference. Whilst the charts show the same relationships of dependence on a ratio of

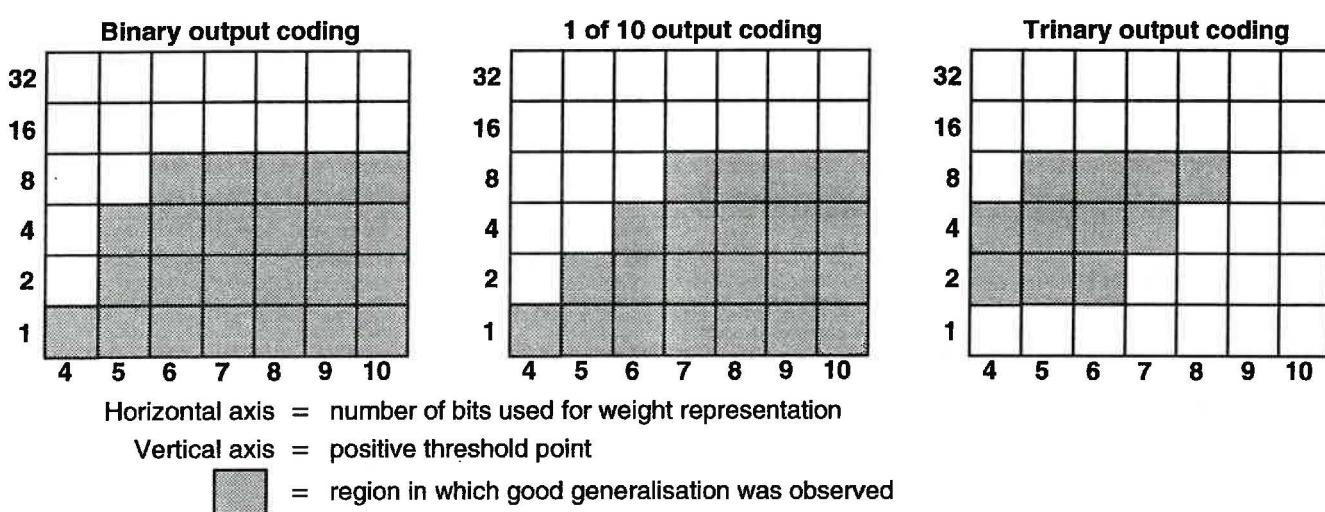


Figure 4. Generalisation as a function of weight magnitude and uncertainty region width.

maximum weight magnitude to uncertainty region width, there exists also an upper limit on the uncertainty region width above which generalisation becomes progressively poorer. In this region, it is not the percentage of correctly classified noisy input patterns that decreases but the percentage of incorrect mappings to other known codes increasing that defines a poor generalisation capability.

4.3. Yield Improvement

It was previously mentioned that one of the expected advantages of the ternary activation function was the improvement of generalisation yield for 1 of x type output codings. The improvement is possible since, provided none of the output values in the code traverse the uncertainty region from an activation of -1 to +1 (or vice versa), the digit can be recognised by the highest activation value in the output vector. To verify this expectation, the percentage of output vectors for which none of the elements traversed the uncertainty region was added to the correctly mapped outputs, the results being shown in Figure 5. The result was a significantly improved generalisation yield for low weight sizes, an advantage for hardware implementation purposes. It should be noted that the condition in which all the output vector elements took the value 0 was not accounted for in this analysis and is probably responsible for the apparent high yield associated with wide uncertainty regions for which its occurrence becomes more probable.

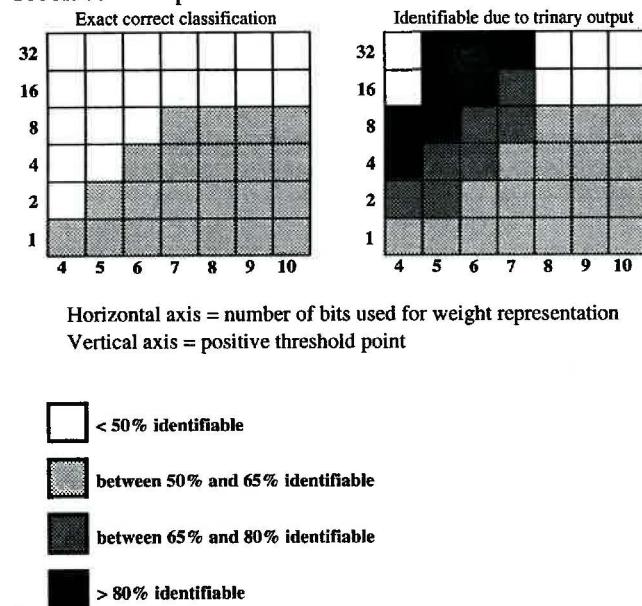


Figure 5. Yield improvement on 1 of x output codings.

5. Conclusion

In conclusion, an adaptation of the back-propagation algorithm which results in a set of integer weight values and uses a ternary activation function has been presented. The results of preliminary investigations on the effectiveness of the training algorithm as a function of the number of allowable weight values and the width of the uncertainty region of the ternary activation function show that the direction is worth pursuing as a means of improving the implementability of ANN's as VLSI parallel

processor arrays. Furthermore, investigation of the use of the ternary activation function as a means of improving generalisation yield for 1 of x type output representations has shown this to be a useful property of the ternary activation function.

In retrospect, the digit recognition problem assigned for evaluating the training algorithm was quite harsh due to the low number of elements in the input vector. Further investigation should include testing the algorithm on larger scale problems in combination with comparisons of the generalisation capabilities of digital networks using a binary activation function and the networks generated by the primary training algorithm.

6. References

- [1] Treleaven, P., Pacheco, M. and Vallasco, M., "VLSI Architectures for Neural Networks", *IEEE Micro*, Vol. 9, No. 6, pp. 9-27, December 1989.
- [2] Goser, K., Hilleringmann, U., Rueckert, U. and Schumacher, K., "VLSI Technologies for Artificial Neural Networks", *IEEE Micro*, Vol. 9, No. 6, pp. 28-44, December 1989.
- [3] Masaki, A., Hirai, Y. and Yamada, M., "Neural Networks in CMOS: a Case Study", *IEEE Circuits and Devices Magazine*, Vol. 6, Issue 4, pp. 12-17, July 1990.
- [4] Caianiello, E.R., "Is There a Silicon Way to Intelligence?", *IEEE Micro*, Vol. 9, No. 6, pp. 75-76, December 1989.
- [5] Stevenson, M., Winter, R., and Widrow, B., "Sensitivity of layered Neural Networks to Errors in Weights", in *Proc. 1990 IJCNN*, Vol. 1, pp. 337-340, 1990.
- [6] Shoemaker, P.A., Carlin, M.J. and Shimabukuro, R.L., "Back Propagation Learning With Trinary Quantization of Weight Updates", *Neural Networks*, Vol. 4, pp. 231-241, 1991.
- [7] Nakayama, K., Inomata, S. and Takeuchi, Y., "A Digital Multilayer Neural Network With Limited Binary Expressions", in *Proc. 1990 IJCNN*, Vol. 2, pp. 587-592, 1990.
- [8] Marchesi, M., Benvenuto, N., Orlandi, G., Piazza, F. and Uncini, A., "Design of Multi-Layer Neural Networks With Powers-of-Two Weights" in *Proc. 1990 IEEE International Symposium on Circuits and Systems*, Vol. 4, pp. 2951-2954, 1990.
- [9] Salam, F.M.A., Choi, M., and Wang, Y., "Artificial Neural Nets in MOS Silicon" in *Artificial Neural Networks and Statistical Pattern Recognition: Old and New Connections* L.K. Sethi and A.K. Jain (Editors), B.V., Elsevier Science Publishers, 1991, pp. 243-270.
- [10] McLelland, J.L., and Rumelhart, D.A., "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", Cambridge, M.A., MIT Press, Vol. 1.
- [11] Foo, S.Y., Anderson, L.R. and Takefuji, Y., "Analog Components for the VLSI of Neural Networks", *IEEE Circuits and Devices Magazine*, Vol. 6, No. 4, pp. 18-26, July, 1990.

A Comprehensive Synthesis Strategy for Conflict Resolutions in Distributed Expert Systems

Minjie Zhang and Chengqi Zhang

Department of Mathematics, Statistics and Computing Science
University of New England, Armidale, N.S.W. 2351, Australia
[{minjie,chengqi }@neumann.une.edu.au](mailto:{minjie,chengqi}@neumann.une.edu.au)

ABSTRACT

In this paper, a comprehensive synthesis strategy for conflict resolutions in distributed expert systems is proposed. It consists of four steps: (a) if the range of uncertainties of solutions are not in [0, 1], the heterogeneous transformation functions are used to transform the uncertainties of propositions from that range to the range of [0, 1]; (b) if the conflict degree is great, the cluster strategy is used to classify the uncertainties into several subsets to reduce the conflict degree; (c) for each subset, if there are more than one uncertainty value, the synthesis strategy for inconsistency is used to obtain the final uncertainty among uncertainty values; and (d) if there are more than one subset, the synthesis strategy for contradiction is used to obtain the final uncertainty among subsets. In each step, values for some important parameters have been selected. The innovations of this strategy are that (a) it can handle quite a wide range of conflicts in distributed expert systems, and (b) it is feasible to use because the values of some important parameters have been selected.

1 Introduction

1.1 Overview

What is conflict? *Conflict* is a pervasive and inevitable aspect of life [1]. Generally speaking, there are three kinds of conflict in human society. These are: belief conflict, resource conflict, and goal conflict [13]. *Belief conflict* appears when cooperating members obtain different results or different uncertainties to the same proposition. For example, when two scientists predict a potential earthquake in a certain area, and both scientists take a geological point of view, one may believe that there is more likely to be a class 7 potential earthquake in the future while the other scientist may insist that a potential earthquake will less likely be class 7. *Resource conflict* occurs when two or more people want to use a given resource which cannot handle both of their demands simultaneously, e.g. two doctors want to use the same "CT machine" at the same time. *Goal conflict* happens when one person has goals that cannot be achieved if other people's goals are to be realized. For instance, when two engineers design a house, the distribution engineer may design large windows on the sun-facing front of a house in order to get more natural light in the house, but the electrical engineer may offer a proposal with small windows in the sun-facing front of the house in order to reduce cooling costs in the summer.

Like human society, the same kinds of conflicts may occur when expert systems (ESs) cooperate together. Belief conflicts are very common conflicts in distributed expert systems (DESs) because most DESs concentrate on getting solutions to problems [11].

Obviously, it is not good for DESs to offer many solutions to the same problem. The *aim of conflict*

resolution in DESs is mainly concerned with deciding what is the final solution based on many solutions.

Because conflict is inevitable in DESs, conflict resolution is a necessary and important aspect of any distributed expert system (DES). Alternatively we can say that a good conflict resolution strategy is a necessary condition for a good DES.

However, it is difficult to design a good conflict resolution strategy in DESs because conflicts are very complex problems. Usually, conflicts occur because the conflicting parties have inconsistent results, insufficient resources, or incompatible goals.

1.2 Conflict analysis in DESs

In DESs, if more than one expert system (ES) is used to solve the same problem, the results may be different. There are roughly three cases which are as follows [6].

Case 1: Absolute conflict.

For the same proposition p , some ESs get the conclusion $p = \text{true}$ while others get the conclusion $p = \text{false}$.

Case 2: Knowing - ignoring.

For the same proposition p , some ESs get the conclusion $p = \text{true}$ or false , but some others cannot get any conclusion.

Case 3: Different values for attributes.

One of the important attributes for an ES is uncertainty. Sometimes, although for the same proposition p , ESs may get different uncertainties, e.g. ES_1 says that the uncertainty is x_1 while ES_2 says the uncertainty is x_2 . If $x_1 \neq x_2$, there is a conflict. We define $|x_1 - x_2|$ as the *conflict degree*.

Suppose we use the EMYCIN [7] inexact reasoning model to represent the uncertainties. The range of uncertainties is from $[-1, 1]$. If the conclusion is $p = \text{true}$, the uncertainty is 1. If the conclusion is $p = \text{false}$, the uncertainty is -1. If the conclusion is unknown, the uncertainty is 0. So if $x_1 = 1$ and $x_2 = -1$ or vice versa, case 3 degenerates to case 1 (absolute conflict). If $x_1 = -1$ or 1 and $x_2 = 0$ or vice versa, case 3 degenerates to case 2 (knowing – ignoring). Therefore, absolute conflict and knowing – ignoring can be represented as the special cases of different values for attributes. In other words, the different solutions from different ESs can be represented as the same solution with different uncertainties. Conflict resolution in this case is how to synthesize these different uncertainties reasonably.

For example, suppose there are three ESs (e.g. ES_1, ES_2, ES_3) to decide the identity of the organism for a specific patient. ES_1 says that it is pseudomonas with uncertainty 0.36 and proteus with uncertainty -0.9, ES_2 says that it is pseudomonas with uncertainty 0.5 and serratia with uncertainty 0.4, and ES_3 says that it is serratia with uncertainty 0.1 and proteus with uncertainty 0.85. Because ES_1 doesn't mention serratia, we believe that ES_1 has no idea about it. We can represent this unknown by using uncertainty 0 in the EMYCIN model. Then the above solutions are represented in table 1.

	Pseudomonas	Serratia	Proteus
ES_1	0.36	0	-0.9
ES_2	0.5	0.4	0
ES_3	0	0.1	0.85

Table 1: The uncertainties for each attribute value obtained by the ESs.

The purpose of the conflict resolution here is how to decide the final uncertainty of pseudomonas from 0.36, 0.5, and 0, or the final uncertainty of serratia from 0, 0.4, and 0.1, and so on.

1.3 Different cases of belief conflicts

Suppose there are n ESs in a DES, the range of uncertainty of ESs is $[-1, 1]$ and the set of uncertainties given by different ESs is $\{x_i\}$, where $i = 1, 2, \dots, n$.

Belief conflicts can be divided into two cases:

Case 1: Inconsistency

All of the ESs in a DES believe that the proposition is partially true (or partially false). The difference is their uncertainties. This situation can be described as: $\forall x_i, x_i > 0$, which means the proposition is partially true, (or $\forall x_i, x_i \leq 0$, which means the proposition is partially false), where 0 represents unknown in the EMYCIN inexact reasoning model [7].

Case 2: Contradiction

The opinions of all ESs in a DES are different. Some of the ESs believe the proposition is partially true but others believe not. We can represent this case in the following way: $\exists x_j \in \{x_i\}, x_j < 0$ and $\exists x_k \in \{x_i\}, x_k > 0$.

In the following sections, we will propose and analyze a comprehensive synthesis strategy for conflict resolution to solve both inconsistency case and contradiction case. In section 2, we outline the principle of the synthesis strategy. In sections 3, 4, 5, and 6, we describe individual part of this strategy respectively. The related work is discussed in section 7 and conclusion is in the last section.

2 The principle of a comprehensive synthesis strategy

2.1 The problem description

Suppose there are n ESs in a DES to evaluate the values of an attribute of an object, (e.g. what is the identity of the organism for a specific patient), the solution for ES_i can be represented as

$$(<\text{object}><\text{attribute}> (V_1 CF_{i1} A_i) (V_2 CF_{i2} A_i) \dots (V_m CF_{im} A_i)) \quad 2.1$$

where V_j represents j th possible value, CF_{ij} represents the uncertainty for j th value from ES_i , A_i represents the authority for ES_i , and m indicates that there are exactly m possible values for this attribute of the object. For example, there are exactly 6 possible values for the face-up of a dice.

From the synthesis point of view, all ESs are concerned with the same attribute of an object. So we will omit the attribute of an object in the representation. Here is the representation of m possible values with uncertainties from n ESs.

$$\begin{array}{|c|c|} \hline & (V_1 CF_{11} A_1)(V_2 CF_{12} A_1)\dots(V_m CF_{1m} A_1) \\ & (V_1 CF_{21} A_2)(V_2 CF_{22} A_2)\dots(V_m CF_{2m} A_2) \\ & \dots \\ & (V_1 CF_{n1} A_n)(V_2 CF_{n2} A_n)\dots(V_m CF_{nm} A_n) \\ \hline & 2.2 \\ \end{array}$$

The synthesis strategy is responsible for obtaining final uncertainties $(V_1 CF_{*1} A_*)(V_2 CF_{*2} A_*) \dots (V_m CF_{*m} A_*)$ based on matrix 2.2.

2.2 The principle of the strategy

This strategy is mainly used to solve belief conflicts in DESs. It includes four steps:

- (a) If the range of uncertainties of solutions is not in $[0, 1]$, the uncertainties of propositions are transformed from that range to the range of $[0, 1]$ by using the heterogeneous transformation functions [8]. For example, if an ES uses the EMYCIN model, the range of $[-1, 1]$ should be transformed into the range of $[0, 1]$.

- (b) If the conflict degree is great, the cluster strategy [6] is used to classify the uncertainties into several subsets. In each subset, the conflict degree should satisfy a certain requirement. At least, it should fall in the case of inconsistency (recall subsection 1.3).
- (c) For each subset, the *synthesis strategy for inconsistency* is used to obtain the final uncertainty among uncertainty values if there is more than one uncertainty value in a subset of uncertainties.
- (d) If there is more than one subset, the *synthesis strategy for contradiction* is used to obtain the final uncertainty for the whole set of uncertainties.

In the steps (c) and (d), both uncertainties of propositions from different ESs and the authorities for each ES are considered.

3 The Heterogeneous Transformation of Uncertainties

In DESs, the transformation of uncertainties of a proposition between different inexact reasoning models is one of the fundamental problems for cooperative problem solving.

It has two roles [10] for heterogeneous transformation:

- (a) If ESs in a DES use different inexact reasoning models, it is necessary to transform the uncertainties of propositions from different models to a single model when ESs cooperate. For example, if one ES uses the EMYCIN model, the range of uncertainties is in $[-1, 1]$ and another ES uses the probability model, the range of uncertainties is in $[0, 1]$. The transformation function should be used to transform the uncertainties of propositions from the EMYCIN model to the probability model ($[-1, 1] \rightarrow [0, 1]$) or vice versa ($[0, 1] \rightarrow [-1, 1]$).
- (b) If the range of uncertainties in some ESs is not in $[0, 1]$, the transformation function should be used to transform the uncertainties to the range of $[0, 1]$, because it is assumed that the synthesis strategy works on the range of $[0, 1]$.

It is very easy to define many transformation functions which can transform uncertainties from one range to another range. Questions here are:

- (a) what are the criteria for judging a good transformation function and
- (b) how to construct a transformation function which is closer to the criteria than other transformation functions.

In [12], Zhang verified that the four popular inexact reasoning models (EMYCIN [7], PROSPECTOR [2],

MYCIN and EVIDENCE THEORY [5]) are semigroup structures. Based on such conclusion, Zhang proposed that homomorphic transformation is the principal criterion for a transformation function. He also defined the transformation functions between any two models among the EMYCIN, PROSPECTOR, and MYCIN models.

Here is a transformation function [8] from the EMYCIN model to the PROSPECTOR model [8] (the range of uncertainties in the PROSPECTOR model is $[0, 1]$).

$$F(x) = \begin{cases} p_0 * (1 + x) & x \leq 0 \text{ } \& p_0 < 0.2 \\ \frac{p_0 * (1 - p_0 + p_0 * x)}{(1 - x) * (1 - 2p_0) + p_0} & x > 0 \text{ } \& p_0 < 0.2 \\ p_0(1 + 0.6x - 0.4x^2) & x \leq 0 \text{ } \& p_0 \geq 0.2 \\ \frac{p_0^2(1 - z)}{(1 - 2p_0)z + p_0^2} & x > 0 \text{ } \& p_0 \geq 0.2 \end{cases}$$

where $z = p_0 * (1 - 0.6x - 0.4x^2)$ and p_0 is the prior probability for a specific proposition in the PROSPECTOR model. The argument x is an uncertainty in the EMYCIN model with the range of $[-1, 1]$. $F(x)$ is an uncertainty in the PROSPECTOR model with the range of $[0, 1]$ corresponding to x in the EMYCIN model. For example, if a proposition P is false, $x = -1$ in the EMYCIN model and $F(-1) = 0$ in the PROSPECTOR model. If a proposition P is unknown, $x = 0$ in the EMYCIN model and $F(0) = p_0$ in the PROSPECTOR model. From now on, we will use p_0 to represent the unknown of a proposition and uncertainty range will be in $[0, 1]$.

This transformation function satisfies the criterion of homomorphic transformation criterion [8].

For example, suppose there are ten ESs working in the EMYCIN model in a DES. Table 2 shows their uncertainties from the EMYCIN model. Table 3 shows the results of the transformation of their uncertainties from the EMYCIN model to the PROSPECTOR model. Here we let $p_0 = 0.2$.

EMYCIN	Pseudomonas	Serratia	Proteus
ES_1	-0.8	0.9	-0.9
ES_2	-0.6	0.8	-0.8
ES_3	-0.4	0.6	0.0
ES_4	-0.2	0.4	-0.1
ES_5	0.0	0.2	-0.2
ES_6	0.2	0.0	-0.2
ES_7	0.4	-0.2	0.7
ES_8	0.6	-0.4	0.7
ES_9	0.8	-0.6	0.2
ES_{10}	0.9	-0.8	0.2

Table 2: The uncertainties in EMYCIN model

4 Cluster Strategy

Suppose there are n ESs in a DES. For a proposition, n ESs will produce a set of n uncertainties $\{x_i\}, i = 1, 2, \dots, n$. Firstly the set $\{x_i\}$ is divided

PROSPECTORS	Pseudomonas	Serratia	Proteus
ES_1	0.05	0.69	0.03
ES_2	0.10	0.53	0.05
ES_3	0.14	0.36	0.20
ES_4	0.17	0.28	0.19
ES_5	0.20	0.23	0.17
ES_6	0.23	0.20	0.17
ES_7	0.28	0.17	0.43
ES_8	0.36	0.14	0.43
ES_9	0.53	0.10	0.23
ES_{10}	0.69	0.05	0.23

Table 3: The transformation of uncertainties from EMYCIN to PROSPECTOR

into two subsets $\{x_j\}$ and $\{x_k\}$, where $\forall_j, x_j \leq p_0$ and $\forall_k, x_k > p_0$, (p_0 is the uncertainty to represent unknown). In each subset, the conflict would be the case of inconsistency (recall subsection 1.3). Secondly, for any one subset $\{x_j\}$ or $\{x_k\}$, if the conflict degree is great, i.e. $\exists l, m, |x_l - x_m| > c_1$ ($c_1 \leq p_0$), the cluster strategy [6] is used to further divide the subset into several small subsets again.

The concept of cluster strategy [6] is to divide the uncertainty set $\{x_j\}$ (or $\{x_k\}$) into subsets $\{g_i\}, i = 1, \dots, m$. So we have

- (1) $\bigcup_{i=1}^m \{g_i\} = \{x_j\}$,
- (2) $\{g_i\} \cap \{g_k\} = \emptyset$, for $i \neq k$, and
- (3) $|x_{i_1} - x_{i_2}| < c_2$
and $\left| \frac{x_{i_2} - x_{i_1}}{x_{i_3} - x_{i_4}} \right| < c_3$, $\forall j_1, j_2, i_1, i_2, i_3, i_4$, such that $j_1 \neq j_2$, $x_{i_1}, x_{i_2}, x_{i_3} \in g_{j_1}$ and $x_{i_4} \in g_{j_2}$ are valid.

Note: we suppose that there are n elements in $\{x_j\}$ and $j \leq n$. We can use the same method to divide the uncertainty set $\{x_k\}$ into subsets.

Here $c_2 \leq p_0$ and c_3 is the parameter used to adjust the number of subsets. We still use the same example as in section 3 to illustrate the cluster algorithm. In this example, let $c_2 = p_0 = 0.2$, $c_3 = 2$. After cluster, the first column has three subsets. They are $subset_1 = \{0.05, 0.10, 0.14, 0.17, 0.2\}$, $subset_2 = \{0.23, 0.28, 0.36\}$, and $subset_3 = \{0.53, 0.69\}$. The second column consists of $subset_1 = \{0.69, 0.53\}$, $subset_2 = \{0.36, 0.28, 0.23\}$, and $subset_3 = \{0.2, 0.17, 0.14, 0.10, 0.05\}$. The third column is divided into $subset_1 = \{0.03, 0.05, 0.20, 0.19, 0.17, 0.17\}$, and $subset_2 = \{0.43, 0.43, 0.23, 0.23\}$.

The fact is, after cluster, different columns in the uncertainty matrix (recall table 3) may have different subsets. For example, after cluster, column 1 is divided into subsets $\{ES_1, ES_2, ES_3, ES_4, ES_5\}$, $\{ES_6, ES_7, ES_8\}$ and $\{ES_9, ES_{10}\}$, column 2 is divided into $\{ES_1, ES_2\}$, $\{ES_3, ES_4, ES_5\}$, $\{ES_6, ES_7, ES_8, ES_9, ES_{10}\}$ and column 3 is divided into $\{ES_1, ES_2, ES_3, ES_4, ES_5, ES_6\}$ and $\{ES_7, ES_8, ES_9, ES_{10}\}$. In this case, what is the result of cluster based on all columns? The simple

method is to find the intersection of these three groups of subsets. In the above example, the intersection will be $\{ES_1, ES_2\} \cap \{ES_3, ES_4, ES_5\} \cap \{ES_6\} \cap \{ES_7, ES_8\} \cap \{ES_9, ES_{10}\}$. The worst case is that the intersection will be $\{ES_1\} \cap \{ES_2\} \cap \{ES_3\} \cap \{ES_4\} \cap \{ES_5\} \cap \{ES_6\} \cap \{ES_7\} \cap \{ES_8\} \cap \{ES_9\} \cap \{ES_{10}\}$.

Therefore, the real result of cluster is shown in table 4.

subset	ES	Pseudomonas	Serratia	Proteus
$subset_1$	ES_1	0.05	0.69	0.03
	ES_2	0.10	0.53	0.05
$subset_2$	ES_3	0.14	0.36	0.20
	ES_4	0.17	0.28	0.19
$subset_3$	ES_5	0.2	0.23	0.17
	ES_6	0.23	0.20	0.17
$subset_4$	ES_7	0.28	0.17	0.43
	ES_8	0.36	0.14	0.43
$subset_5$	ES_9	0.53	0.10	0.23
	ES_{10}	0.69	0.05	0.23

Table 4: The result of cluster

5 Synthesis Strategy for Inconsistency

In matrix 2.2, for the one column l , if $\forall i$, CF_{il} belongs to the interval $[0, P_0]$, or $\forall i$, CF_{il} belongs to interval $[P_0, 1]$, then we say the solutions of all ESSs satisfy the “consistency treatment condition”, so the consistency treatment strategy can be used (see section 5.1). After cluster the n uncertainties in one column l is divided into subsets. The uncertainties in every subset satisfy “consistency treatment condition”.

In each subset, a strategy is needed to synthesize these uncertainties to obtain a final uncertainty if there is more than one uncertainty in a subset.

In this section, a synthesis strategy is described which includes:

- (a) consistency treatments for modifying ESSs' uncertainties; and
- (b) the synthesis of uncertainties.

5.1 Consistency treatments

The purpose of consistency treatments is to adjust the uncertainties of all ESSs bases on the influences between them. In this paper, we introduce a reciprocal influence strategy for the consistency treatments [4].

The principal aim of the reciprocal influence strategy is to emphasize the role of all ESSs and the influences among them. The basic ideas are:

- (a) quality requirement: If $CF_i > CF_j$, then $CF'_i < CF_i$ after CF_i is influenced by CF_j , where CF'_i represents the CF_i after adjusting.

(b) quantity requirement: The contribution of CF_i' plays major role to the CF_i' . The influences of other ESs on CF_i' is minor.

Based on these requirements, CF_i' is given by the following formula:

$$CF_i' = CF_i + \alpha * \sum_{j=1}^k \frac{A_j}{A_j + A_i} * (1 - p_0 - |CF_j - CF_i|) * (CF_j - CF_i) \quad 5.3$$

5.3

In formula 5.3, CF_i is the original uncertainty, α is a constant which is used to adjust the degree of influence of other ESs on ES_i , and A_i is the authority of ES_i . Because the influence from other ESs should be minor, we take $\alpha = \frac{1}{k}$, where k is the number of ESs in a subset. Formula 5.3 can work properly only when CF_i satisfies the consistency treatment condition. Otherwise, it may violate the requirement for quality. For example, if $CF_1 = 0.95, CF_2 = 0.1, p_0 = 0.2, \alpha = 0.5, A_1 = A_2 = 0.9$, then $CF_1' = 0.967$, which is bigger than CF_1 .

Here is the example of the consistency treatments. We use the results of the example of table 4 in section 4 to test the consistency treatments. The results after adjusting for five subsets are shown in table 5. We still take $p_0 = 0.2$.

subset	α	ES	Autho- rity	Pseu- omonas	Serr- atia	Prot- eus
$subset_1$	0.5	ES_1	0.6	0.06	0.67	0.03
		ES_2	0.5	0.09	0.56	0.05
$subset_2$	0.3	ES_3	0.7	0.15	0.34	0.20
		ES_4	0.8	0.17	0.28	0.19
		ES_5	0.6	0.19	0.25	0.18
$subset_3$	1	ES_6	0.7	0.23	0.20	0.17
$subset_4$	0.5	ES_7	0.5	0.30	0.16	0.43
		ES_8	0.8	0.35	0.14	0.43
$subset_5$	0.5	ES_9	0.8	0.55	0.09	0.23
		ES_{10}	0.7	0.66	0.06	0.23

Table 5: The result of consistency treatment

5.2 Synthesis of uncertainties

After adjusting uncertainty values using the above strategy, a set of uncertainties are still different. The purpose of synthesis is to combine these adjusted uncertainties in some appropriate way to obtain an uncertainty value to present a common view of the ESs in a subset.

Suppose that FCF denotes a final uncertainty value, $MEAN$ is the mean value of all uncertainties (CFs) in a subset, and $UNIFORMITY$ is the deviation of uncertainty values, the FCF will depend on both $MEAN$ and $UNIFORMITY$ of uncertainties. One method of calculating $MEAN$ and $UNIFORMITY$ is as follows:

$$MEAN = \sum_{i=1}^k CF_i' * \frac{A_i}{\sum_{j=1}^k A_j} \quad 4.4$$

$$UNIFORMITY = \sum_{i=1}^k |CF_i' - MEAN| * \frac{A_i}{\sum_{j=1}^k A_j} \quad 4.5$$

Note: $UNIFORMITY=0$ if all CF_i' are same. Otherwise $UNIFORMITY > 0$.

The formula of calculating FCF is:

$$FCF = \begin{cases} \gamma * MEAN - \beta_1 * UNIFORMITY & \text{if } MEAN \geq p_0 \\ \gamma * MEAN + \beta_2 * UNIFORMITY & \text{if } MEAN < p_0 \end{cases}$$

where γ , β_1 , and β_2 are constants.

Reasons for us to use both $MEAN$ and $UNIFORMITY$ to calculate the final uncertainty value are: (a) $MEAN$ is the principal part of the final uncertainty value; and (b) for the same $MEAN$, different $UNIFORMITY$ should slightly affect the final uncertainty value.

If $MEAN \geq p_0$, FCF represents partial truth. In this case, the $UNIFORMITY$ will reduce the degree of partial truth. So we use minus between $MEAN$ and $UNIFORMITY$. In contrast, if $MEAN < p_0$, that means FCF represents partially false. In this case, the $UNIFORMITY$ will still reduce the degree of partially false. So we use plus between $MEAN$ and $UNIFORMITY$. Take particular note, if an uncertainty is less than p_0 , the smaller the uncertainty, the stronger the degree of falsity. The extreme cases are that if $CF = 0$, it is the strongest falsity, but if $CF = p_0$, it is the weakest falsity.

Now we decide the coefficient γ, β_1 and β_2 . If all ESs give the same uncertainty, i.e. $UNIFORMITY = 0$, then FCF should be equal to the $MEAN$. Therefore $\gamma = 1$ is reasonable. (Note, if the entropy method is chosen to calculate $UNIFORMITY$, $\gamma < 1$ [9]).

If $MEAN \geq p_0$ (i.e. each uncertainty is greater than p_0), we believe FCF should be bigger than p_0 , because there is no reason why the final uncertainty value represents partially false when each uncertainty represents partially true. So $MEAN - \beta_1 * UNIFORMITY \geq p_0$, i.e., $0 < \beta_1 \leq \frac{MEAN - p_0}{UNIFORMITY}$. The upper limit of β_1 will depend on the maximum of the $UNIFORMITY$. In a subset with k uncertainties, if $\frac{k}{2}$ uncertainties are equal to p_0 and other $\frac{k}{2}$ uncertainties are equal to 1, the $UNIFORMITY$ of k uncertainties is maximum. In this case, $MEAN$ is $\frac{1+p_0}{2}$, $UNIFORMITY$ is $(\frac{k}{2}(1 - MEAN) + \frac{k}{2}(MEAN - p_0))/k = (1 - \frac{1+p_0}{2}) = \frac{1-p_0}{2}$, $\beta_1 \leq (\frac{1-p_0}{2} - p_0)/(\frac{1-p_0}{2}) = 1$. If $\beta_1 = 0$, the $UNIFORMITY$ has no affect on FCF . If $\beta_1 = 1$, the $UNIFORMITY$ has the biggest affect on FCF . We choose $\beta_1 = \frac{1}{2}$ so that $UNIFORMITY$ has a moderate affect on FCF . Using the same idea, we can calculate the upper limit $\beta_2 = 1$ too. So we choose $\beta_2 = \frac{1}{2}$ again.

After cluster, if more than one subset exists, the synthesis strategy for contradiction is used to calculate the final uncertainty value for all subsets (see next section). In this case, we should decide a new authority for each subset. The formula is the geometric mean: $FA = \sqrt[k]{\prod_{i=1}^k A_i}$ where FA means a new authority

for a subset and k is the number of uncertainties in the subset. We know the geometric mean is less than the arithmetic mean. The reason that we select the geometric mean is that FCF for a subset is weaker than $MEAN$ (which is evaluated by the arithmetic mean) because of the effect from $UNIFORMITY$. Thus FA for a subset should be weaker than the arithmetic mean of authorities.

We continue using the results of examples in section 5 to illustrate this synthesis strategy.

	<i>MEAN</i>	<i>UNIFORMITY</i>	<i>FA</i>	<i>FCF</i>
<i>subset₁</i>	0.0736	0.0149	0.5477	0.0811
<i>subset₂</i>	0.1690	0.0127	0.6952	0.1754
<i>subset₃</i>	0.2300	0.0000	0.7000	0.2300
<i>subset₄</i>	0.3308	0.0237	0.6325	0.3189
<i>subset₅</i>	0.6013	0.0548	0.7483	0.5740

Table 6: The synthesis results for Pseuomonas

	<i>MEAN</i>	<i>UNIFORMITY</i>	<i>FA</i>	<i>FCF</i>
<i>subset₁</i>	0.6200	0.0545	0.5477	0.5927
<i>subset₂</i>	0.2914	0.0324	0.6952	0.2752
<i>subset₃</i>	0.2000	0.0000	0.7000	0.2000
<i>subset₄</i>	0.1477	0.0095	0.6325	0.1524
<i>subset₅</i>	0.076	0.0149	0.7483	0.0835

Table 7: The synthesis results for Serratia

	<i>MEAN</i>	<i>UNIFORMITY</i>	<i>FA</i>	<i>FCF</i>
<i>subset₁</i>	0.0391	0.0099	0.5477	0.0410
<i>subset₂</i>	0.1905	0.0063	0.6952	0.1937
<i>subset₃</i>	0.1700	0.0000	0.7000	0.1700
<i>subset₄</i>	0.4300	0.0000	0.6325	0.4300
<i>subset₅</i>	0.2300	0.0000	0.7483	0.2300

Table 8: The synthesis results for Proteus

After combining tables 6, 7 and 8, (keep FCF and FA only), it is shown in table 9.

6 Synthesis Strategy for contradiction

From section 5, we know that the synthesis strategy for inconsistency can calculate the final uncertainty value for each subset after cluster. However, if there is more than one subset after cluster, the synthesis strategy for contradiction is needed to calculate the final uncertainty value among subsets from each final uncertainty value for each subset. In this case, evidential theory [5] is used to form the synthesis strategy for contradiction.

Suppose S is a finite set, $S = \{s_1, s_2, \dots, s_m\}$ where s_1, s_2, \dots, s_m are propositions. 2^S denotes the set of all subsets of S . Suppose μ_k and μ_l are two basic support functions over the same set 2^S . $\mu_k(s_i)$ and $\mu_l(s_i)$ are used to measure the uncertainty of proposition s_i by ES_k and ES_l respectively.

The synthesis function is defined as follows:

	Pseuomonas	Serratia	Proteus	<i>FA</i>
<i>subset₁</i>	0.08	0.59	0.04	0.55
<i>subset₂</i>	0.18	0.28	0.19	0.70
<i>subset₃</i>	0.23	0.20	0.17	0.70
<i>subset₄</i>	0.32	0.15	0.43	0.63
<i>subset₅</i>	0.57	0.08	0.23	0.75

Table 9: Three values from five subsets

$$\mu(s_i) = \frac{\mu_k(s_i)\mu_l(s_i) + \mu_k(s_i)\mu_l(s) + \mu_k(s)\mu_l(s_i)}{\mu_k(s)\mu_l(s) + \sum_{i=1}^m (\mu_k(s_i)\mu_l(s_i) + \mu_k(s_i)\mu_l(s) + \mu_k(s)\mu_l(s_i))} \quad 6.1$$

If the denominator of formula 6.1 is 0, formula 6.1 is undefined. This means that μ_k and μ_l are in absolute conflict. In this case, the synthesis strategy cannot be used and other strategies are suggested to be used to solve the conflict.

In the above functions, $\mu_l(s_i)$ and $\mu_l(s)$ are given by

$$\mu_l(s_i) = \frac{\mu'_l(s_i)}{\mu'_l(s) + \sum_{i=1}^m \mu'_l(s_i)} \quad 6.2$$

$$\mu_l(s) = \frac{\mu'_l(s)}{\mu'_l(s) + \sum_{i=1}^m \mu'_l(s_i)} \quad 6.3$$

$$\mu'_l(s_i) = \begin{cases} \mu''_l(s_i) * FA_l & \text{if } \mu''_l(s_i) < p_0 \\ \mu''_l(s_i) & \text{otherwise} \end{cases} \quad 6.4$$

$$\mu'_l(s) = \begin{cases} \mu''_l(s) * FA_l & \text{if } \mu''_l(s) < p_0 \\ \mu''_l(s) & \text{otherwise} \end{cases} \quad 6.5$$

$$\mu''_l(s_i) = F_1(F_2(CF_{li}) * FA_l) \quad 6.6$$

$$\mu''_l(s) = F_1(F_2(1 - \sum_{i=1}^m CF_{li}) * FA_l) \quad 6.7$$

where CF_{li} is the uncertainty value for the i th possible value from the l th subset after cluster, FA_l is the authority for the l th subset. F_1 is the transformation function from the EMYCIN model to the PRESPECTOR model and F_2 is the inverse function of F_1 (recall section 3). The same method can be used to define $\mu_k(s_i)$ and $\mu_k(s)$.

Why do we define $\mu'_l(s_i)$ in formula 6.6 this way? The reasons are as follows:

Case 1: If $FA_l = 1$, i.e. full authority, then adjusting $\mu''_l(s_i)$ should be the same as CF_{li} . Formula 6.6 works as

$$\mu'_l(s_i) = F_1(F_2(CF_{li}) * 1) = F_1(F_2(CF_{li})) = CF_{li}$$

because F_1 is the inverse function of F_2 .

Case 2: If $FA_l = 0$, i.e. not any authority, then adjusting $\mu''_l(s_i)$ should be unknown. Formula 6.6 should be as $\mu'_l(s_i) = F_1(F_2(CF_{li}) * 0) = F_1(0) = p_0$ because 0 in the EMYCIN model represents an unknown and function F_1 transfers the unknown in the EMYCIN model to the unknown in the PROSPECTOR model.

Case 3: The bigger the authority, the stronger the degree of truth. For example, suppose FA_l is 0.9, FA_k is 0.1, $CF_{l_i} = 0.8$, $CF_{k_i} = 0.8$ and $p_0 = 0.2$, then $\mu_l''(s_i) = F_1(F_2(0.8) * 0.9) = 0.60$ and $\mu_k''(s_i) = F_1(F_2(0.8) * 0.1) = 0.21$. So $\mu_l''(s_i) > \mu_k''(s_i)$;

Case 4: The bigger the authority, the stronger the degree of falsity. For example, suppose $FA_l = 0.9$, $FA_k = 0.1$, $CF_{l_i} = 0.1$, $CF_{k_i} = 0.1$, and $p_0 = 0.2$, (note: if an uncertainty is less than p_0 , then the uncertainties represent partially false), then $\mu_l''(s_i) = F_1(F_2(0.1) * 0.9) = 0.11$ and $\mu_k''(s_i) = F_1(F_2(0.1) * 0.1) = 0.19$. So $\mu_l''(s_i) < \mu_k''(s_i)$.

It should be noted that synthesis strategy for inconsistency is concerned with one column only when it synthesizes different uncertainties in one column (recall section 5). However, synthesis strategy for contradiction is concerned with the whole uncertainty matrix when it synthesizes different uncertainties in one column.

Each time, two uncertainty values can be synthesized using formula 6.1. If there are n ESs in a DES, after applying formula 6.1 $n - 1$ times, the final uncertainty value for the proposition s_i is obtained. Note, each time a new authority is needed for further synthesis. The formula can be $NFA = \sqrt{W_l \cdot FA_l^{W_l} \cdot FA_k^{W_k}}$, where W_l is the weight for l th subset and W_k is the weight for k th subset. The weight W is measured by the number of elements in a subset divided by the number of elements in the whole set. The formula for the new weight should be $NW = W_l + W_k$.

We take Table 9 as an example.

After using formula 6.1 four times, three uncertainties for three values from five subsets are obtained which are shown in table 10.

	Pseudomonas	Serratia	Proteus
Final CF	0.507	0.168	0.323

Table 10: The final result of uncertainties for three values.

7 Related Works

During recent years, conflict resolution has been a critical capacity required for coordination in DESs. The synthesis method is considered to be a good way to solve belief conflicts in DESs [6].

In 1985, Khan [3] first proposed an uncertainty management strategy to solve belief conflicts in DESs. In this strategy, the contributions are:

- (a) cooperative ESs influence each other and show cooperative behavior by modifying their values of uncertainties; and
- (b) the final decision of a certain proposition is based on not only the average of uncertainties of the

propositions but also the uniformity among ESs about the proposition.

This strategy still had several limitations:

- (a) It assumes the range of uncertainties for all ESs is in $[0, 1]$. To those ESs whose uncertainties take from $[-1, 1]$ or other ranges, such as the EMYCIN model, this strategy doesn't work.
- (b) When it synthesizes the uncertainties, it does not take the authority of each ES into account.
- (c) When the conflict degree is great, it doesn't work well.
- (d) In this strategy, there are several uncertain parameters to decide the contributions of different factors. Khan didn't give a clue about how to set them. If the parameters are set improperly, this strategy will be less useful.

In 1990, Zhang [9] introduced a homomorphic transformation method to realize the heterogeneous transformation of the uncertainty of a proposition between different inexact reasoning models. This method improved Khan's strategy to work in the EMYCIN model. So it overcame the first limitation only of Khan's strategy. Zhang also proposed a new method for calculating the average value of uncertainties and uniformity corresponding to the new range of uncertainties.

Recently, Liu [4] has done further research and introduced the influence of the authorities of ESs. So this strategy overcomes the second limitation only of Khan's strategy. Liu also introduced a new method for calculating the average value of uncertainties and uniformity considering both uncertainties of propositions and authorities of ESs.

Lu [6] introduced a cluster strategy to further classify all uncertainties in a DES into several subsets to reduce the conflict degree. It overcomes the third limitation only of Khan's strategy.

Summarizing, Khan [3] proposed the basic framework of the synthesis strategy in DESs. Zhang [9], Liu [4], and Lu [6] improved the framework by each overcoming one limitation. However, no one strategy can work well to solve a wide range of conflicts. The key problems remaining in synthesis strategies in previous research are:

- (a) how to allow these strategies to be really useful, because each strategy can only solve problems in special conditions;
- (b) how to find clues to determine some important parameters in these strategies. Otherwise these strategies cannot solve real conflicts. This is a very difficult and important problem. It has not been solved in previous researches.

In our new comprehensive strategy, we have:

- (a) combined the advantages of the above strategies to solve a wide range of conflicts; and

- (b) given a guideline for setting some important parameters in the strategy and they are verified.

8 Conclusion

In this paper, a comprehensive synthesis strategy is proposed to solve belief conflicts in DESs. This strategy can solve a wide range of conflicts in DESs because it overcomes the limitations of other strategies. This strategy is feasible because the values for some important parameters have been selected and verified.

The limitation of this strategy is that it can't solve absolute conflict. Furthermore, this strategy will be used in real DESs to analyze its effects.

References

- [1] M. Deutsch, Conflict and its Resolution. In C. G. Smith (Ed) *Conflict Resolution: Contributions of the Behavioral Science*, University of Notre Dame Press, London, 1971.
- [2] R. Duda, P. Hart & N. Nilsson, Subjective Bayesian Method for Rule-based Inference System. *AFIPS*, 45, pp. 1075-1082, 1976.
- [3] N. A. Khan & R. Jain, Uncertainty Management in a Distributed Knowledge Based System. In *Proceedings of 5th International Joint Conference on Artificial Intelligence*, Los Angeles, pp. 318-320, 1985.
- [4] D. Liu, F. Zheng, Z. Ma & Q. Shi, Conflict Resolution in Multi-ES cooperation Systems. *Journal of Computer Science and Technology*, Vol. 7, No. 2, pp. 144-152, 1992.
- [5] G. Shafer, A Mathematical Theory of Evidence. *Princeton University Press*, Princeton and London, 1976.
- [6] R. Lu, Expert Union: United Service of Distributed Expert Systems, *Technical Report 85-3, Department of Mathematical Sciences, University of Minnesota, Duluth, Minnesota*, 1985.
- [7] W. Van Melle, A Domain-independent System that Aids in Constructing Knowledge-Based Consultation Programs. *Ph.D. Dissertation, Report STAN-CS-80-820, Computer Science Department, Stanford University, CA*, 1980.
- [8] C. Zhang, Homomorphic Transformation of Uncertainties of Propositions among Inexact Reasoning Models. *IEEE transactions on Knowledge and Data Engineering*, Vol. KDE-6, 1994.
- [9] C. Zhang, The Management of Uncertainty in a Distributed Expert System. *AI'90*, edited by C. P. Tsang, World Scientific Publishing Co., Singapore, pp. 313-322, 1990.
- [10] C. Zhang, Cooperation Under Uncertainty in Distributed Expert Systems. *Artificial Intelligence*, Vol. 56, pp. 21-69, 1992.
- [11] C. Zhang & D. Bell, HECODES: A Framework for Heterogeneous Cooperative Distributed Expert System. *International Journal on Data & Knowledge Engineering*, Vol. 6, pp. 251-273, 1991.
- [12] C. Zhang & M. Orlowska, On algebraic Structures of Inexact Reasoning Models, In: G.E. Lasker et al., eds., *Advances in Information Systems Research*, pp. 58-77, 1991.
- [13] M. Zhang & C. Zhang, Negotiation as a Metaphor for Conflict Resolution in Distributed Artificial Intelligence. In *Proceedings of CWC ICIA'93*, Beijing, China, 1993.

Family Law, Natural Language Understanding and AI

W.K. Yeap
*Artificial Intelligence Laboratory,
 Computer and Information Science,
 University of Otago, New Zealand
 e-mail: coscwky@otago.ac.nz*

Abstract

In the Artificial Intelligence Research Laboratory at Otago University, we are interested in fundamental research on the nature of intelligence and in particular, how to develop a machine with original intentionality [12]. Our first step is to develop computational ideas about the humans perceptual processes. Our interests in applied research lie only in areas which overlap with our basic research and/or in areas which allow us to explore related fundamental issues. In this paper, we describe one such area of research jointly undertaken between members of the Laboratory and the Law Faculty. Our basic research goal is to develop a computational theory of natural language understanding and the applied research goal is to develop an intelligent system for use in the area of Family Law.

1 Introduction - Family Law

We describe a research project on natural language understanding and law which was initiated over two years ago in the Artificial Intelligence Research Laboratory at Otago University. A chance meeting with Prof. Sutton¹ (of the Law Faculty) in a tramping trip on the Milford Track introduced us to a peculiar situation in the practice of Family Law in New Zealand.

The Family Protection Act 1955 was drafted to ensure that people distribute their estate fairly upon death. A claim under this Act is brought by one or more plaintiffs who are asking for re-consideration of the distribution of an estate. If the claim is valid, an appropriate change is made to the deceased's will. For each case, the judge's oral history of the family involved and the decision reached is transcribed to produce a law case as a record of the decision (Figure 1²).

Fei Low Chan of Dunedin, retired farmer, died on the 19 January 1987 leaving a last will dated 22 October 1986. That will provided for the appointment of the defendant as trustee, left tools to the deceased's brother Sor Low Chan, provided \$15,000 to be divided equally between the three daughters of the deceased living at the date of death of the survivor of himself and his wife, and the residue to the Middlemarch Hospital Board, with a request that it be used for the purchase of equipment in the coronary care unit at the Middlemarch Hospital. Mr Chan was 64. He and the plaintiff in these proceedings, Eileen Chan, had married in the United Kingdom in 1943. At the time they were both in the Air Force. They returned to New Zealand to the Dunedin area where the deceased had come from. There were three children - Ting born in April 1945, Lei born in December 1946 and Fai born in 1957.

Figure 1. An example of a part of a transcription of a judge's report on the Family Law case.

¹who is now seconded to be the NZ Law Commissioner.

²names and places in the report are changed for obvious reasons.

However, unlike the practice of other areas of law (such as Criminal Law), these records are not made widely available and as such went largely unnoticed by other judges. Consequently each judge deals with a new case with little hindsight from previous judges' decision. One reason for this might be because the factors influencing a judge's decision in each case are often unclear from the report itself, thus making the report less useful. The latter is not due to an omission on the part of the judges when reporting a case but rather that the Family Protection Act operates in an area of law in which there are no strict rules. The judges have to interpret each case according to their own perception of family values and how these should be translated into monetary terms. Such decisions are very difficult to make and quantified, and naturally change over time and place.

By attempting to identify the influencing factors in each case manually, one could develop a case-based reasoning program [5]. But, such a program would be of limited use since, as we pointed out above, these factors and how they affect each case could change over time. Our problem is thus unlike a medical problem; we could retrieve a similar case (like some judges do remember some of them) but what conclusions one should draw from them remains a problem. This, together with our strong interests in working on more fundamental problems [12], leads us to work on a program which, simply put, could read and 'understand' each case. Such a program could then be used to analyse a number of cases to identify the various factors affecting each case and how they relate to the judges' decision. This would allow one to understand how judges' perception of family values have changed over the years and how the judgement itself has also changed. Highlighting the influencing factors in this way could in turn help judges to understand how a new case relates to old ones. Perhaps the program could also be used to predict judges' decision for new cases.

The remainder of this paper describes our work done so far: from the development of a new cognitive theory of natural language understanding program called LINGUIST to the initial development of LAWYER, a text interpretation program for analysing text on family history, and other related programs. The description here is intended to be

brief and non-technical; technical reports of our work are currently in preparation (but see [13]).

2 Natural Language Understanding

At the heart of our problem is the design of an intelligent natural language understanding program. We note that the prevailing cognitive model for understanding natural language since the 1960's has been a syntax-semantic model for which AI researchers have provided various ingenious methods for its physical realization (for recent work, see [2,4]).

However, we are dissatisfied with this basic model³ since its process structure is delimited by studying what language is rather than by considering what needs to be computed from the input and why. This is particularly worrying since implementations of this model offer little support for the theory. The initial syntactic model was inspired mainly by the success in designing compilers for artificial languages and the later incorporation of semantic knowledge is done with little or no underlying theory to support its use. Without investigating what possible intermediate steps are involved in the process, one fails to understand natural language. Thus, at best, in Marr's [3] words, current AI "theories" of natural language are a Type 2 theory. That is, their result is merely in the form of a large program.

We are keen to develop a more cognitively based model and the basic idea is to emphasise the use of semantic information at the word level. One major problem in developing a theory of natural language understanding based on semantics lies in not knowing how word meanings are combined to produce the final interpretation.

Early AI attempts have focused on the use of word meanings; for example, representing them in such a way as to be able to determine what to expect from the input [8,10]. Consequently, the interpretation problem simply becomes a matching problem. This is unfortunate for two reasons. First, such complex use of word knowledge may be premature given that one has not yet perceived the whole sentence. Note that while we acknowledge that in some situations this is possible, we argue that the use of such knowledge should be to supplement the basic process rather than, as suggested by these researchers, replace it. Second, if the use of word knowledge is so fundamental to

the process then it is important to know exactly how it should be represented. Most of the representations suggested (such as a script or a template) are overly simplistic. Not surprisingly, the success of these approaches is limited [9].

We developed a more basic algorithm, one which is based on the idea that the meanings of a word read are combined with the possible interpretations obtained so far. In other words, multiple interpretations are constantly being constructed as each word is read [13].

Figure 2 illustrates this basic idea using visual images to capture the interpretations of words. Note that we make no claim as to whether visual images are an essential part of our theory; it is still very much a part of our investigation as to what needs to be made explicit at the word level and how. Recently it was drawn to our attention Pustejovsky's work [6,7] on lexical knowledge representation and it looks promising. We will investigate that shortly. In the current implementation we use a simple frame-like representation for word meanings.

An important step in our algorithm is how meanings of later words are combined with earlier ones. We introduce the idea of focus (or active) images, that is, those to which incoming images could be added. Every sentence has a theme which one could think of as the main image which is always active. We also assume that the most recent image added to it is also active. This then allows the main image to be composed of a hierarchy of images within which there is a path of active images leading to the most recently constructed image. Various rulings, syntactic, commonsense, or otherwise⁴, are used to decide where and how an incoming image is combined with these active images. The result is a set of images, each for a different interpretation of the sentence. Figure 3 shows an example of an output from LINGUIST displayed in a discourse window. Multiple interpretations produced are resolved at the next level, when more contextual information becomes available.

LINGUIST is implemented in CLOS and currently consists of about 120 verbal image classes and 70 rulings. It accepts genitives, comparative clauses, infinitive clauses, imperatives, wh-clauses, parenthetic expressions, conjunctions and others. We have successfully tested the parsing of several ambiguous sentences. For more information, see [13].

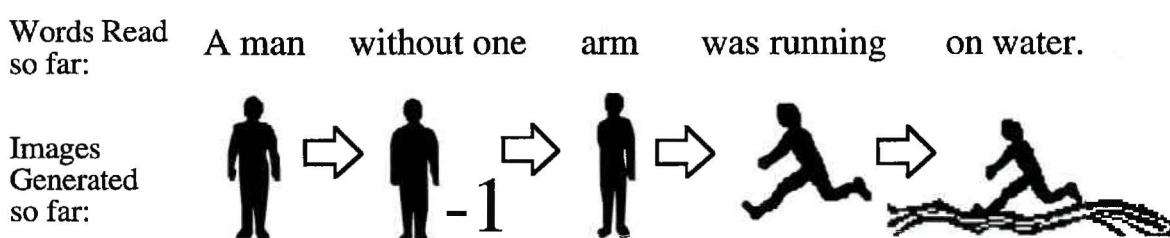


Figure 2. Meanings of a sentence are actively constructed as each word is read from the sentence

³which we also believe that the sentences in each case (see Figure 1) are difficult to parse using the traditional approach.

⁴Many of the rulings currently implemented are syntax rules and we are currently investigating the use of other rulings.

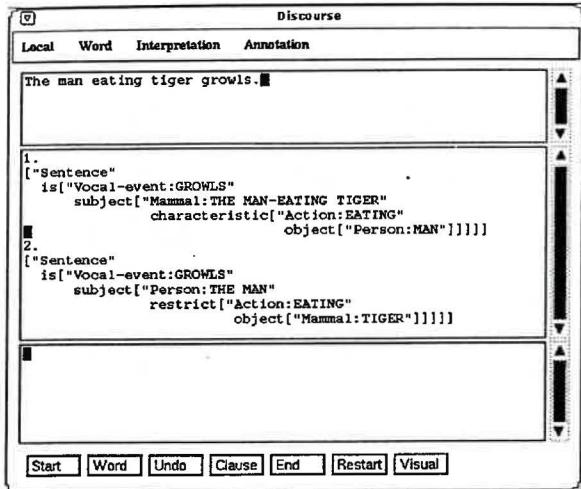


Figure 3. An example output from LINGUIST

3 Text Interpretation

More recently, we have begun to investigate how interpretations of single sentences provided by LINGUIST could be combined. We want to continue with our idea of manipulating word meanings as the primary basis for doing so. To illustrate, consider the following three sentences:

John Picard died on the 1st of April.
He made a will.
He left tools to his brother.

Interpretation of the first sentence is as follows:

[“Event: DIED”
 subject[“Person: JOHN PICARD”]
 when[“Date: 1st of APRIL”]

↓

[*Person
 :id_Num person001
 :name (john picard)
 *:Living (:status False :time (1st of april))]

The interpretation process takes the output from LINGUIST and create instances of important artifacts or people. Note that we are interested in capturing the state of what has already happened but contrary to the expectation-based approach, this is obtained not from filling in slots but from building up a picture of what was said. Thus, we created “John died” not because we expected that a person can die but because it was mentioned in the sentence itself. The ‘*’ indicates that these instances are currently in focus and we will see how this is used later. Interpretation of the next sentence is as follows:

[“Action: MADE”
 subject[“Personal-pro-form: HE”]
 object[“Legal-act: A WILL”]

↓

5using a simplified output from LINGUIST.

[*Person

:Id_Num person001
:Name (john picard)
:Sex male
:Living (:status false :time (1st of april))
*:Possess (will002)]

[*Will

:Id_Num legal-act002
:Quantity 1]

We will not discuss how anaphoric resolution is achieved here but just to point out that it could be tricky, especially with sentences like the following which is not uncommon in these cases:

“Somehow Edward was able to find *his* father’s address in Thailand and *he* wrote to *him*, and *his* father responded and gave *his* consent and *he* came out to the wedding.”

After the interpretation of the second sentence, the focus is now on the word “will” rather than on the event “death”. This has an interesting consequence for the interpretation of the third sentence as shown below:

[“Action: LEFT”

subject[“Personal-pro-form: HE”]
object[“Artifact: TOOLS”]
beneficiary[“Relative: BROTHER”]
relative[“Personal-possessive-pro-form:
HIS”]]

↓

[*Person

:Id_Num person001
:Name (john picard)
:Sex male
:Living (:status false :time (1st of april))
*:Brother (person004)
*:Possess (legal-act002)]

[*Artifact

:Id_Num artifact003
:Name tool
:Quantity some]

[*Will

:Id_Num legal-act002
:Quantity 1
:contend ((artifact003 -> person004))]

[*Person

:Id_Num person004
:Brother (person001)
:Sex male]

LINGUIST identifies ‘left’ as a transfer event as opposed to a motion event and if one ignores what has been said earlier, one could have created a slot “:possess (artifact003)” in the representation for “brother”. But, since the will is in focus and it is a description of what things should be transferred to whom, the interpretation of the third sentence goes into the will.

We are currently working on LAWYER to build up a picture of a family history by manipulating the word meanings and how they appear in the text. The usefulness of this approach will also be tested to see if one can resolve anaphoric references.

4 Related Areas of Interests

We will briefly mention here two related projects that have arisen from our work on LINGUIST. The first is to investigate how well our theory of natural language understanding works for other languages. Does it work for all languages? Is it a suitable framework for work on machine translation? Our initial work is to build a similar program for the Japanese language which we dubbed *J-LINGUIST*. Figure 4 shows an example output.

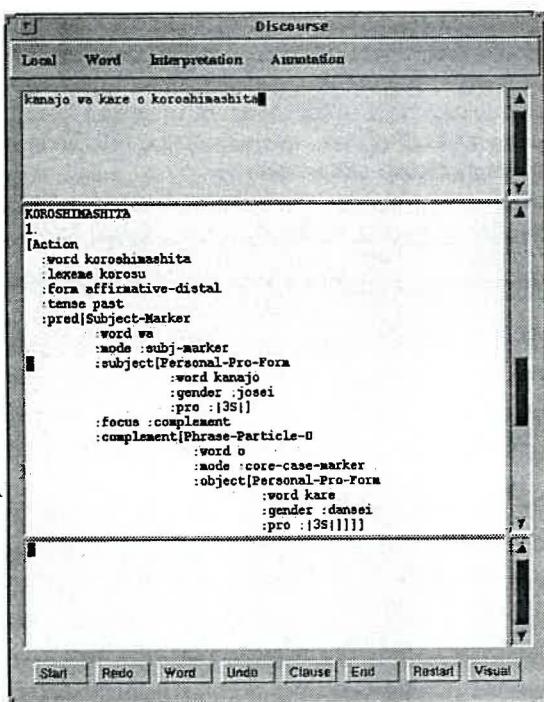


Figure 4. Parsing a Japanese sentence

The second is to investigate how we might be able to provide a visual description of the meaning of a sentence (for similar work in the literature, see [1]). Figure 5 shows our initial attempt to visualise the sentence "Roy is in the park".

5 Concluding Remark

When presented with the opportunity to study a problem in Family Law, we were motivated to develop a computational theory of natural language understanding⁶. Our initial development and testing of the theory looks promising and have also encouraged us to investigate other interesting problems.

⁶Prior to this, we have been working on a computational theory of cognitive maps (see [11,14]).

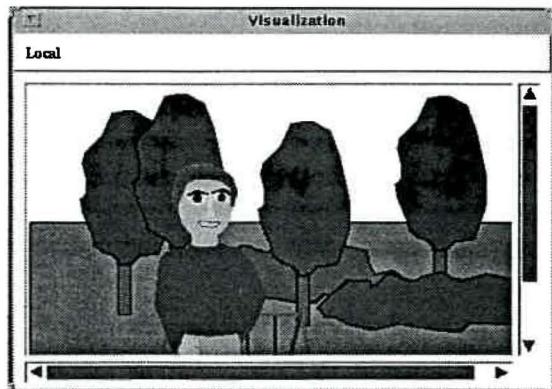


Figure 5. Visualising "Roy is in the park"

6 Acknowledgement

This project was supported by grants from the Law Faculty, Division of Sciences and the Information Science Department. Participants in the project both in the past and the present include: Prof. Sutton, Ms. Peart, Ms. Buckingham (Law), Prof. Sallis (Information Science), Dr. Brugman (English); Mr. Anderson (research assistant). A special thanks to all who have commented on our work and especially to Dr. Jones of Victoria University. My special thanks also to my students who have helped to test my ever-changing ideas of what the theory should be: Matt, Brendon, Lindsay, Mike, Craig, Dale, and Jane.

7 References

- [1] Adorni, G., Manzo, M.D. and Giunchiglia, F. "From Descriptions to Images: What reasoning in between?", Proceedings of the European Conference on AI, 139-148, 1985.
- [2] Barnett, J., Knight, K., Mani, I. and Rich, E. "Knowledge and Natural Language Processing", CACM, 33, 8, 50-71, 1990.
- [3] Marr, D. "Artificial Intelligence - A Personal View", Artificial Intelligence, 9, 37-48, 1977.
- [4] Palmer, M.S., Passonneau, R.J., Weir, C., and Finin, T. "The KERNEL Text Understanding System", Artificial Intelligence, 63, 17-68, 1993.
- [5] Porter, B.W., Bareiss, R. and Holte R.C. "Concept Learning and Heuristic Classification in Weak-Theory Domains", Artificial Intelligence, 45, 229-263, 1990.
- [6] Pustejovsky, J. "The Syntax of Event Structure", Cognition, 41, 47-81, 1991.
- [7] Pustejovsky, J. and Boguraev, B. "Lexical Knowledge Representation and Natural Language Processing", Artificial Intelligence, 63, 193-223, 1993.

[8] Riesbeck, C. "Conceptual Analysis" in: R. Schank, ed., *Conceptual Information Processing* (North-Holland, Amsterdam) 89-147, 1975.

[9] Ritchie, G. "Semantics in Parsing", in: M. King, ed., *Parsing Natural Languages* (Academic Press, London), 199-217, 1983.

[10] Wilks, Y. "Preference Semantics", in E.L. Keenan, ed., *Formal Semantics of Natural Language Processing* (Cambridge University Press), 1975.

[11] Yeap, W.K. "Towards a Computational Theory of Cognitive Maps", *Artificial Intelligence*, 34, 297-360, 1988.

[12] Yeap, W.K. "Emperor AI: Where is Your New Mind?", University of Otago AI Memo AIM-20-92-2, 1992.

[13] Yeap, W.K. and Anderson, R. "Computing Mental Sketches as a Basis for Natural Language Interpretation", University of Otago AI Memo AIM-25-94-1, 1994.

[14] Yeap, W.K. and Handley, C.C. "Four Important Issues in Cognitive Mapping", in J. Meyer and S.W. Wilson, eds., *From Animats to Animats* (MIT Press), 176-183, 1991.

An Introduction to Evolutionary Computation

David B. Fogel
Natural Selection, Inc.
 1591 Calle De Cinco
 La Jolla, CA 92037
 fogel@sunshine.ucsd.edu

Abstract

The impact of evolutionary theory on biology cannot be underestimated. But evolutionary theory extends beyond an ordering principle of life. The fundamental processes in evolution can be simulated on a computer and used for good engineering purpose. Evolutionary computation is the field of research investigating simulations of natural evolution. This paper offers a brief introduction to three main avenues of investigation: Genetic algorithms, evolution strategies, and evolutionary programming. The methods are broadly similar, but each adopts a slightly different level of abstraction. Some simple examples of each technique are offered.

Modeling Evolutionary Optimization

Darwinian evolution provides an essential framework for biology [29]. Evolution can be viewed as an optimization process: Organisms iteratively improve their ability to predict environmental circumstances so as to survive and reproduce. The challenges that individuals and species face are typified by nonlinearities, chaos, random and temporal variation, and other difficulties, and yet natural evolution has invented "organs of extreme perfection" [4] to cope with these challenges. These same general characteristics have also posed significant challenges to conventional optimization algorithms. Capturing the essential aspects of Darwinian evolution in algorithmic form can lead to robust optimization techniques capable of addressing problems that are currently intractable.

The most widely accepted collection of evolutionary theories is the neo-Darwinian paradigm. These arguments assert that the vast majority of the history of life can be accounted for by physical processes operating on and within populations and species [23]. These processes are reproduction, mutation, competition, and selection. Reproduction is an obvious property of extant species. Mutation, in any positively entropic universe, is guaranteed in that errors during replication of genetic information will necessarily occur. Competition is a consequence of expanding populations in a finite resource space. Selection is the inevitable result of competitive replication as species fill the available space. Evolution is then an inescapable result of

these basic interacting physical processes [46].

Evolved individuals and species can be viewed as a duality of their genetic program, the *genotype*, and their expressed behavioral traits, the *phenotype* (Figure 1). The genotype provides a mechanism for the storage of experiential evidence, of historically acquired information. But the resulting phenotypic expression of a genetic structure is generally unpredictable due to the universal effects of *pleiotropy* and *polygeny*. Pleiotropy is the effect that a single

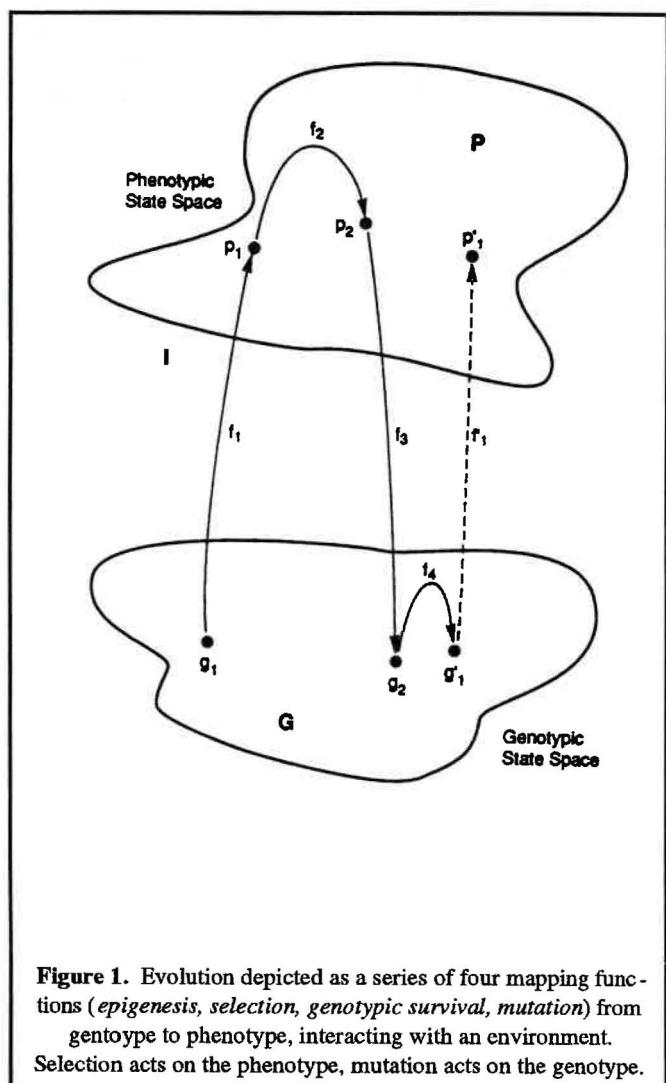


Figure 1. Evolution depicted as a series of four mapping functions (*epigenesis, selection, genotypic survival, mutation*) from genotype to phenotype, interacting with an environment. Selection acts on the phenotype, mutation acts on the genotype.

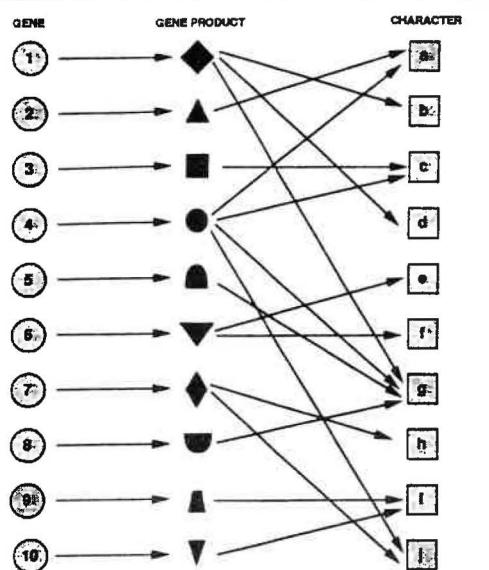


Figure 2. The relationship between the genotype and phenotype is pleiotropic (one gene affects many traits) and polygenic (one trait affected by many genes).

gene can affect multiple phenotypic traits, and polygeny is the effect that a single phenotypic trait can be affected by many genes (Figure 2). Different genetic structures may imply equivalent behaviors and the realized phenotype also depends on the interaction between the organism and its environment.

Selection acts only on the expressed behaviors of individuals and species [29]. This is perhaps best illustrated using the idea of an adaptive topography (response surface) as proposed by Wright [47]. A population of genotypes maps to respective phenotypes which are in turn mapped onto the adaptive topography. The topography indicates the fitness of the individual or population. Evolution probabilistically proceeds up the slopes of the topography, which may contain multiple optima, toward peaks as selection culls inappropriate variants. This process may be equivalently viewed from an inverted position, with selection eliminating those species with greatest predictive error relative to current environmental demands (Figure 3). Evolution then probabilistically descends slopes on the topography searching for troughs.

Viewed in this manner, evolution is an obvious optimization, problem-solving search process. Selection drives phenotypes as close to the optimum as possible, given initial conditions and environmental constraints. But the environment continually changes and species lag behind, evolving toward new optima. Ultimately, suboptimal behaviors must be expected in any dynamic environment, but selection continues to operate regardless of a population's position on the adaptive topography.

This view of evolution leads naturally to two fundamen-

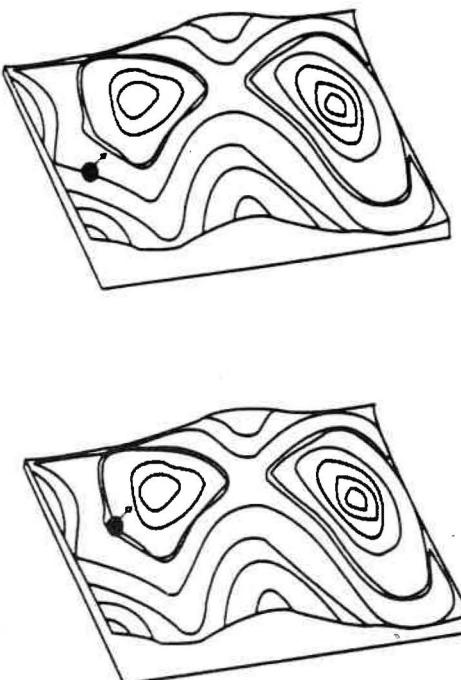


Figure 3. Wright's adaptive topography, inverted. Populations descend the slopes of the topography toward positions of lower behavioral error during evolution.

tally different methods of simulation. The first emphasizes the genotype and the interactions that can be observed at this level. Such procedures have become known as *genetic algorithms*. The second emphasizes the phenotype and suggests that instead of incorporating genetic operators as observed in nature, trial solutions can be perturbed with various operators such that there is a continuous normally distributed variation in the observed behaviors of new trials (or nearly so in discrete problems). Such procedures have become known as *evolutionary algorithms* (although some use this term synonymously with evolutionary computation). Within evolutionary algorithms, attention can be focused on the behavior of individuals, or on the behavior of entire species. These methods are described as *evolution strategies* and *evolutionary programming*, respectively. Each of these methods is described in greater detail in the following sections.

Genetic Algorithms

Many researchers have proposed simulations of genetic systems [3, 20, 24, 36]. These genetic algorithms are typically implemented as follows:

1. A problem to be addressed is defined and captured in

an objective function that indicates the fitness of any potential solution.

2. A population of candidate solutions is initialized subject to applicable constraints. Typically, each solution is coded as a vector \mathbf{x} , termed a chromosome, with elements of the vector being called genes, and alternative values at specific positions termed alleles. Often, these solutions are coded in binary, following suggestions in [24]. If the problem involves continuous-valued optimization then the degree of required precision determines the length of the binary coding.

3. Each chromosome, \mathbf{x}_i , $i = 1, \dots, P$, in the population is decoded into a form appropriate for evaluation and is assigned a fitness score $F(\mathbf{x})$ according to the objective.

4. Each chromosome is assigned a probability for reproduction, p_i , $i = 1, \dots, P$, so that its likelihood of being selected is proportional to its fitness relative to the other chromosomes in the population.

5. According to the assigned probabilities p_i , $i = 1, \dots, P$, a new population of chromosomes is generated by probabilistically selecting solutions from the current population. The selected chromosomes generate "offspring" via the use of specific genetic operators, such as crossover and bit mutation. Crossover operates on two parents and creates two new offspring by selecting a random position along the coding and splicing the section that appears before the selected position in the first solution with the section that appears after the selected position in the second solution, and vice versa. Bit mutation simply offers the possibility of flipping any bit in the coding of a new solution (Figure 4). Typical ranges for the probabilities for crossover and mutation are 0.6-0.95 and 0.001-0.01, respectively [38].

6. The process is halted if a suitable solution has been found, or if the available computing time has expired, otherwise the process continues to step 3, where the new chromosomes are scored and the cycle is repeated.

This general procedure has been used to address many difficult combinatorial optimization problems, including the traveling salesman problem [45], protein secondary and tertiary structure prediction [28], automatic programming [26], and others [19].

There are many issues that must be addressed using a genetic algorithm. For example, the necessity for binary coding has received considerable attention and many researchers have used other representations with success. Selection in proportion to fitness can be problematic for negative objective values, but methods for rescaling the data have been suggested (e.g., fitness based on ranking). The repeated application of crossover and selection tends to homogenize the population, leading potentially to convergence at suboptimal solutions (i.e., *premature convergence*). Specific tricks have been invented to maintain diversity such as only accepting new offspring into a population if they are structurally significantly different from their parents [6]. While the application of genetic algorithms to optimization problems may not be straight-forward, the method has demonstrated potential.

Evolutionary Algorithms

An alternative method for simulating evolution was independently adopted by Rechenberg and Schwefel [33, 39] collaborating in Germany, and L. Fogel [15] in the United States. These models, commonly described as evolution strategies or evolutionary programming, or more broadly as evolutionary algorithms [8, 32], emphasize the behavioral link between parents and offspring, or between reproductive populations, rather than the genetic link. When applied to real-valued function optimization, the most simple method is implemented as follows:

1. The problem is defined as finding the real-valued n -dimensional vector \mathbf{x} that is associated with the extremum of a function $F(\mathbf{x})$: $\mathbf{R}^n \rightarrow \mathbf{R}$. Without loss of generality, let the procedure be implemented as a minimization process.

2. An initial population of parent vectors, \mathbf{x}_i , $i = 1, \dots, P$, is selected at random from a feasible range in each dimension. The distribution of initial trials is typically uniform.

3. An offspring vector, \mathbf{x}'_i , $i = 1, \dots, P$, is created from each parent \mathbf{x}_i by adding a zero mean Gaussian random variable with preselected standard deviation to each component



Figure 4. In genetic algorithms, crossover exchanges segments of two strings, while bit mutation alters a single component of one string.

of \mathbf{x} .

4. Selection then determines which of these vectors to maintain by comparing the errors $F(\mathbf{x}_i)$ and $F(\mathbf{x}'_i)$, $i = 1, \dots, P$. The P vectors that possess the least error become the new parents for the next generation.

5. The process of generating new trials and selecting those with least error continues until a sufficient solution is reached or the available computation is exhausted.

In this model, each component of a trial solution is viewed as a behavioral trait, either of an individual or species, not as a gene. A genetic source for these phenotypic traits is presumed but the nature of the linkage is not detailed. It is assumed that whatever genetic transformations occur, the resulting change in each behavior will follow a Gaussian distribution with zero mean difference and some standard deviation. Specific genetic alterations can affect many phenotypic characteristics due to pleiotropy and polygeny. It is therefore appropriate to simultaneously vary all of the components of a parent in the creation of a new offspring.

The original efforts in evolution strategies examined the preceding algorithm but focused on a single parent, single offspring search. This was termed a (1+1)-ES. Extensions to multiple parents and offspring were offered in [40], and described as $(\mu+\lambda)$ -ES and (μ,λ) -ES. In the former, μ parents are used to create λ offspring and the best μ solutions from all $\mu+\lambda$ are selected as new parents. In the latter, μ parents again give rise to λ offspring, but the new parents are selected only from the λ offspring.

Operations have also been added to adapt the distribution of normal variation around each parent [40]. These self-adaptive mechanisms are associated with each parent, so that each solution carries parameter values to be optimized, but also incorporates information on how to change itself when creating new offspring. This mechanism allows for trials to be adaptively aligned with arbitrary contours on a response surface, rather than being forced to search mainly along coordinate axes (Figure 5). Extensions have also been made to include recombinatory operators on these self-adaptive parameters. This is reasonable in evolution strategies because the solutions are abstracted individuals. Under

evolutionary programming, however, no recombination operators are applied because the solutions are abstracted species and no sexual recombination occurs by definition.

The original evolutionary programming approach was similar to that of evolution strategies, but involved a more complex problem: Creating artificial intelligence. Fogel [16, 17] proposed that intelligent behavior requires the composite ability to predict one's environment coupled with a translation of the predictions into a suitable response in light of a given goal. Environments were simulated as growing sequences from an alphabet of finite symbols and finite state machines were evolved to predict these sequences.

The prediction problem is a sequence of static optimization problems in which the adaptive topography (fitness function) is time-varying. The process can be easily extended to abstract situations in which the payoffs for individual behaviors depend not only on an extrinsic payoff function, but also on the behavior of other individuals in the population (e.g., the iterated prisoner's dilemma) [9].

Evolutionary programming has also been applied to real-valued continuous optimization problems, such as training neural networks [1, 30], and automatic control [10, 43], and is virtually equivalent in many cases to the procedures of evolution strategies, despite their independent development. The extension to using self-adaptive mechanisms for creating offspring was offered in [13, 14], although with slightly different procedures. Comparisons between the various self-adaptive methods is an open area of research; initial comparisons [37] appear to favor the use of the methods proposed in [40]. Further investigations in evolution strategies and evolutionary programming can be found in [5, 42].

Examples

The following examples illustrate the basic operations of the main variations of evolutionary computation. Each technique is applied to a very simple problem. Extensions of these applications are discussed below.

Genetic Algorithms

Consider the problem of finding the string of 100 bits

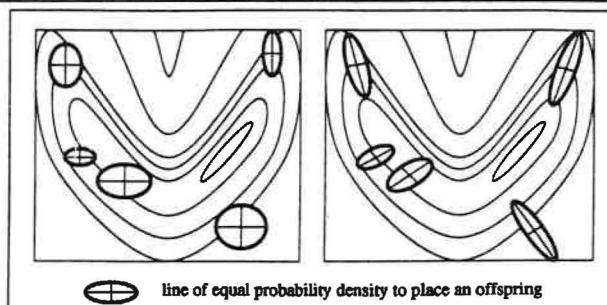


Figure 5. Distributing trials in relation to the contours of the topography. By incorporating self-adaptive mechanisms into the evolutionary search, it can learn to distribute trials so as to increase the likelihood for discovering improved solutions.

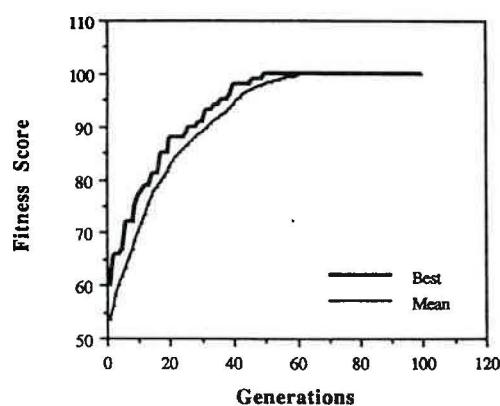


Figure 6. The rate of optimization under a genetic algorithm for the problem of finding a string with maximum bit sum.

{0,1} such that the sum of the bits in the string is maximized. The objective function may be written as

$$F(\mathbf{x}) = \sum x_i, i = 1, \dots, 100.$$

An initial population of binary strings is taken at random with each bit having an equal probability of taking the value 1 or 0. Let the population size be 100 parent strings. Each string is evaluated in light of $F(\mathbf{x})$ and receives a fitness score, which in this case is simply the number of 1's in the string. A new population is created by probabilistically selecting strings from the initial population in proportion to their relative fitness. Thus a string with a fitness of 60 would be twice as likely to be selected as a string with a fitness of 30. The new population is created from the selected parents using the genetic operators of crossover and mutation. Let the crossover rate be 0.8 and the bit mutation rate be 0.01. The process of evaluating the new offspring and selecting new parents is then continued. Figure 6 indicates the rate of optimization of the best score in the population and the mean of all parents' scores at each iteration. The genetic algorithm quickly converges on strings with all components set to 1.

Evolution Strategies

Consider the problem of finding the minimum of the quadratic function of three variables:

$$F(\mathbf{x}) = \sum x_i^2, i = 1, 2, 3.$$

Rather than use any binary transformation on the evolving solutions, the real values are manipulated directly. Let the initial population consist of 30 parents, with each component varying in accordance with a uniform distribution over [-5.12, 5.12] (after [6]). Let one offspring be created from

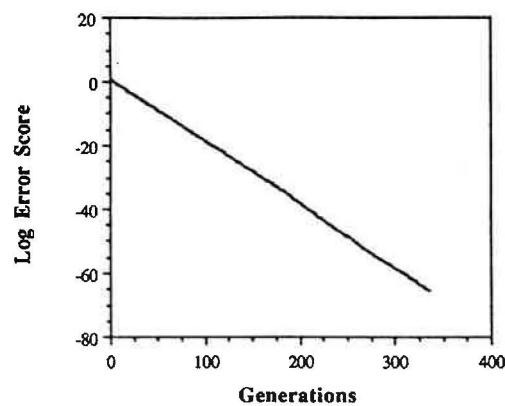


Figure 7. The rate of optimization under evolution strategies for minimizing a three-dimensional quadratic function.

each parent by adding a Gaussian random variable with mean zero and variance equal to the error score of the parent divided by the square of the number of dimensions to each component (Rechenberg [34] indicated that this is nearly optimal for the quadratic function at hand). Let selection simply retain the best 30 vectors in the population of parents and offspring. Figure 7 indicates the rate of optimization of the best vector in the population as a function of the number of generations. The process rapidly converges close to the unique global optimum.

Evolutionary programming

For ease of description, attention is given to a problem considered in Fogel et al. [17]. Consider the use of finite state machines (Figure 8) to represent the logic underlying a sequence of observed symbols. Let the environment be the nonstationary sequence generated by classifying each of the increasing integers as being prime (symbol 1) or nonprime (symbol 0). Thus the environment consists of the sequence 01101010001..., where each symbol depicts the primeness of the positive integers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ..., respectively. Let the payoff function be all-or-none, that is, given a sequence of known symbols, one point is awarded if the finite state machine predicts the next symbol in the sequence correctly and no points are awarded if the prediction is incorrect. Further, let a penalty for complexity of 0.01 multiplied by the number of states of the machine be imposed on the machine's score, so as to maintain parsimonious machines.

Fogel et al. [17] used five finite state machines as the initial population. The initial machines possessed five states. Each machine generated a single offspring through simple mutation, with equal probabilities of either (1) adding a state, (2) deleting a state, (3) changing the start state, (4) changing an output symbol, or (5) changing a next-state transition. The parents and offspring were evaluated in light

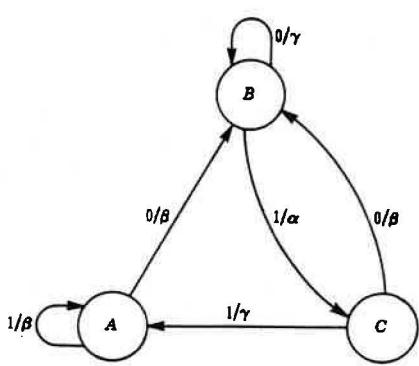


Figure 8. A finite state machine.

of how well they fit the observed sequence of 1's and 0's and 10 iterations of mutation and selection were conducted before the best finite state machine in the population was used to predict the next, as yet unseen, symbol. The accuracy of the prediction was recorded and then the new symbol was added to the observed sequence and the process iterated.

Figure 9 shows the cumulative percentage of correct predictions in the first 200 symbols. After the initial fluctuation (due to the small sample size), the prediction score increased to 78 percent at the 115th symbol and then essentially remained constant until the 200th prediction. At this point, the best machine possessed four states. At the 201st prediction, the best machine possessed three states, and at the 202nd prediction, the best machine possessed only one state with both output symbols being 0. After 719 symbols, the process was halted with the cumulative percentage of correct predictions reaching 81.9 percent. The asymptotic worth of this machine would be 100 percent correct because the prime numbers become increasingly infrequent and the machine continues to predict "nonprime." Additional experiments in Fogel et al. [17] considered forcing the payoff to reflect the rarity of the event, thereby increasing the value for correctly predicting prime numbers (see [17, 12] for details).

Extensions

Each of these techniques has been extended beyond its original formulation. For example, as genetic algorithms often prematurely converge to suboptimal solutions and the provided precision is restricted by the length of the coding, Schraudolph and Belew [38] invented an essentially iterative method of rescaling the available range using *dynamic parameter encoding*. When the population is believed to have converged to a collection of similar solutions, the

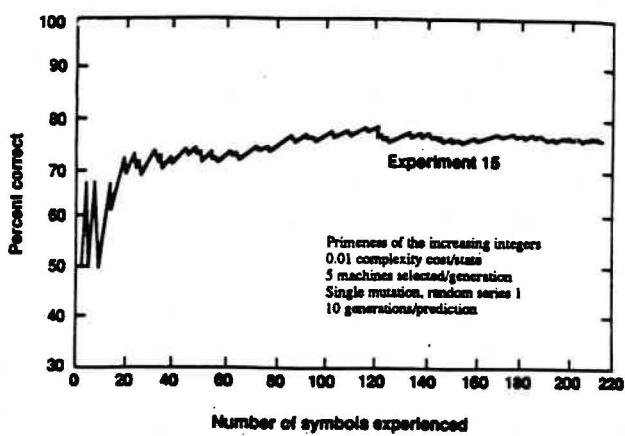


Figure 9. The cumulative score for predicting prime numbers from the experiment offered in [17].

minimum and maximum values in each dimension are recalculated based on the converged population and the search is reinitialized within that new range. Thus the eventual precision of the algorithm is not dependent on the initial coding length.

Other efforts in genetic algorithms have used real-valued representations as opposed to binary coding and have indicated such efforts to be more efficient and effective at discovering improved solutions [31]. In addition, Koza [26] has used genetic algorithms to evolve S-expressions as computer programs. Recent efforts in this area of genetic programming can be found in Kinnear [25].

Further, evolution strategies and evolutionary programming have been applied to a wide variety of combinatorial and discrete optimization problems, including the design and training of neural networks [12], fuzzy control systems [22], and image processing [21]. For a more detailed introduction to these efforts, the reader is encouraged to review [2, 7, 8, 11, 18, 27, 35, 41, 44].

Discussion

Simulated evolution has a long history. Similar ideas and implementations have been invented numerous times independently. There are currently three main lines of investigation: Genetic algorithms, evolution strategies, and evolutionary programming. These methods share many similarities. Each maintains a population of trial solutions, imposes random changes to these solutions, and incorporates the use of selection to determine which solutions to maintain into future generations and which to cull from the pool of trials. But these methods also have important differences. Genetic algorithms emphasize models of genetic operators as observed in nature, such as crossing over and point muta-

tion, and apply these to abstracted chromosomes. Evolution strategies and evolutionary programming emphasize mutational transformations that maintain behavioral linkage between each parent and its offspring, respectively at the level of the individual or population. Recombination may be appropriately applied to individuals, but is not applicable for species.

No model can be a complete description of a true system. All models are incomplete. But each of the above models has been demonstrated to be a useful method for addressing difficult optimization problems. The greatest potential for the application of evolutionary computation will come from implementation on parallel machines, for evolution is an inherently parallel process. Although such efforts will undoubtedly be of practical utility, the ultimate advancement of the field will always rely on the careful observation and abstraction of the natural process of evolution.

Acknowledgments

The author would like to thank L.J. Fogel for his comments and criticisms of this paper, and Y. Attikiouzel for his encouragement and support.

References

- [1] Angeline, P.J., Saunders, G.M., and Pollack, J.B., "An Evolutionary Algorithm that Constructs Recurrent Neural Networks," *IEEE Trans. Neural Networks*, Vol. 5:1, 1994, pp. 54-65.
- [2] Bäck, T., and Hoffmeister, F., "Basic Aspects of Evolution Strategies," *Stat. and Comp.*, Vol. 4:2, 1994, pp. 51-64.
- [3] Bremermann, H.J. "Optimization Through Evolution and Recombination," In *Self-Organizing Systems*, edited by M.C. Yovits, G.T. Jacobi, and G.D. Goldstine, Washington D.C.: Spartan Books, 1962, pp. 93-106.
- [4] Darwin, C., *On the Origin of Species*, London: Murray, 1859, Ch. 6.
- [5] Davidor, Y., Schwefel, H.-P., and Männer, R., *Parallel Problem Solving from Nature*, 3, New York: Springer-Verlag, 1994, in press.
- [6] De Jong, K.A., "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," Ph.D. Diss., Univ. of Michigan, Ann Arbor, 1975.
- [7] De Jong, K.A., "Genetic Algorithms: A 25 Year Perspective," In *Computational Intelligence: Imitating Life*, edited by J.M. Zurada, R.J. Marks, and C.J. Robinson, Piscataway, NJ: IEEE Press, 1994, pp. 125-134.
- [8] Fogel, D.B., "On the Philosophical Differences between Evolutionary Algorithms and Genetic Algorithms," *Proc. of 2nd Ann. Conf. on Evol. Prog.*, edited by D.B. Fogel and W. Atmar, La Jolla, CA: Evolutionary Programming Society, 1993, pp. 23-29.
- [9] Fogel, D.B., "Evolving Behaviors in the Iterated Prisoner's Dilemma," *Evol. Comp.*, Vol. 1:1, 1993, pp. 77-97.
- [10] Fogel, D.B., "Applying Evolutionary Programming to Selected Control Problems," *Comp. Math. Applic.*, Vol. 27, 1994, pp. 89-104.
- [11] Fogel, D.B., "Evolutionary Programming: An Introduction and Some Current Directions," *Statistics and Computing*, Vol. 4:2, 1994, pp. 113-129.
- [12] Fogel, D.B., *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, Piscataway, NJ: IEEE Press, 1995, in press.
- [13] Fogel, D.B., Fogel, L.J., and Atmar, J.W., "Meta-Evolutionary Programming," *Proc. 25th Asilomar Conference on Signals, Systems, and Computers*, edited by R.R. Chen, San Jose, CA: Maple Press, 1991, pp. 540-545.
- [14] Fogel, D.B., Fogel, L.J., Atmar, W., and Fogel, G.B., "Hierarchic Methods of Evolutionary Programming," *Proc. 1st Ann. Conf. on Evol. Prog.*, edited by D.B. Fogel and W. Atmar, La Jolla, CA: Evolutionary Programming Society, 1992, pp. 175-182.
- [15] Fogel, L.J., "Autonomous Automata," *Industrial Research*, Vol. 4, 1962, pp. 14-19.
- [16] Fogel, L.J., "On the Organization of Intellect," Ph.D. Diss., UCLA, 1964.
- [17] Fogel, L.J., Owens, A.J., Walsh, M.J. *Artificial Intelligence Through Simulated Evolution*, New York: John Wiley, 1966.
- [18] Fogel, L.J., "Evolutionary Programming in Perspective: The Top-Down View," In *Computational Intelligence: Imitating Life*, edited by J.M. Zurada, R.J. Marks, and C.J. Robinson, Piscataway, NJ: IEEE Press, 1994, pp. 135-146.

- [19] Forrest, S. (ed.), *Proc. of the Fifth Intern. Conf. on Genetic Algorithms*, San Mateo, CA: Morgan Kaufmann, 1993.
- [20] Fraser, A.S., "Simulation of Genetic Systems by Automatic Digital Computer," *Australian J. of Biol. Sci.*, Vol. 10, 1957, pp. 484-491.
- [21] Füger, H., Stein, G., and Stilla, U., "Multi-Population Evolution Strategies for Structural Image Analysis," *Proc. of 1st IEEE Conf. on Evolutionary Computation*, Piscataway, NJ: IEEE Press, 1994, pp. 229-234.
- [22] Haffner, S.B., and Sebald, A.V., "Computer-Aided Design of Fuzzy HVAC Controllers," *Proc. of 2nd Ann. Conf. on Evol. Prog.*, edited by D.B. Fogel and W. Atmar, La Jolla, CA: Evolutionary Programming Society, 1993, pp. 98-107.
- [23] Hoffman, A., *Arguments on Evolution*, Cambridge, MA: Oxford, 1989, p. 39.
- [24] Holland, J.H. *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI: Univ. of Michigan Press, 1975.
- [25] Kinnear, K.E. (ed.) *Advances in Genetic Programming*, Cambridge, MA: MIT Press, 1994.
- [26] Koza, J.R., *Genetic Programming*, Cambridge, MA: MIT Press, 1992.
- [27] Koza, J.R., "Genetic Programming as a Means for Programming Computers by Natural Selection," *Stat. and Comp.*, Vol. 4:2, 1994, pp. 87-112.
- [28] Lucasius, C., "Toward Genetic Algorithm Methodology in Chemometrics," Ph.D. Diss., Univ. Nijmegen, The Netherlands, 1993.
- [29] Mayr, E., *Toward a New Philosophy of Biology: Observations of an Evolutionist*, Harvard: Belknap, 1988, pp. 477-478.
- [30] McDonnell, J.R., and Waagen, D., "Evolving Recurrent Perceptrons for Time-Series Modeling," *IEEE Trans. Neural Networks*, Vol. 5:1, 1994, pp. 24-38.
- [31] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, New York: Springer-Verlag, 1992.
- [32] Mühlenbein, H., "Evolution in Time and Space — The Parallel Genetic Algorithm," *Foundations of Genetic Algorithms*, edited by G.J.E. Rawlins, San Mateo, CA: Morgan Kaufmann, 1992, p. 316-337.
- [33] Rechenberg, I., "Cybernetic Solution Path of an Experimental Problem," Royal Aircraft Establishment, Lib. Trans. 1122, Farnborough, 1965.
- [34] Rechenberg, I., *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*, Stuttgart: Frommann-Holzboog Verlag, 1973.
- [35] Rechenberg, I., "Evolution Strategy," In *Computational Intelligence: Imitating Life*, edited by J.M. Zurada, R.J. Marks, and C.J. Robinson, Piscataway, NJ: IEEE Press, 1994, pp. 147-159..
- [36] Reed, J., Toombs, R., and Barricelli, N.A., "Simulation of Biological Evolution and Machine Learning," *J. Theo. Biol.*, Vol. 17, 1967, pp. 319-342.
- [37] Saravanan, N., and Fogel, D.B., "Learning Strategy Parameters in Evolutionary Programming: An Empirical Study," *Proc. of the 3rd Ann. Conf. on Evol. Prog.*, edited by A.V. Sebald and L.J. Fogel, River Edge, NJ: World Scientific Publishers, 1994, in press.
- [38] Schraudolph, N.N., and Belew, R.K., "Dynamic Parameter Encoding in Genetic Algorithms," *Machine Learning*, Vol. 9, 1992, pp. 9-21.
- [39] Schwefel, H.-P., "Kybernetische Evolution als Strategie der Experimentellen Forschung in der Stromungstechnik," Diploma Thesis, Tech. Univ. of Berlin, 1965.
- [40] Schwefel, H.-P., *Numerical Optimization of Computer Models*, Chichester, U.K.: John Wiley, 1981.
- [41] Schwefel, H.-P., "On the Evolution of Evolutionary Computation," In *Computational Intelligence: Imitating Life*, edited by J.M. Zurada, R.J. Marks, and C.J. Robinson, Piscataway, NJ: IEEE Press, 1994, pp. 116-124.
- [42] Sebald, A.V., and Fogel, L.J. (editors) *Proceedings of the Third Annual Conference on Evolutionary Programming*, River Edge, NJ: World Scientific Publishing, 1994.
- [43] Sebald, A.V., and Schlenzig, J., "Minimax Design of Neural Net Controllers for Highly Uncertain Plants," *IEEE Trans. Neural Networks*, Vol. 5:1, 1994, pp. 73-82.
- [44] Whitley, D., "A Genetic Algorithm Tutorial," *Stat. and Comp.*, Vol. 4:2, 1994, pp. 65-86.

[45] Whitley, D., Starkweather, T., and Shaner, D., "The Traveling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination," In *Handbook of Genetic Algorithms*, edited by L. Davis, New York: Van Nostrand Reinhold, 1991, pp. 350-372.

[46] Wooldridge, D.E., *The Mechanical Man*, New York: McGraw-Hill, 1968, p. 25.

[47] Wright, S., "The Roles of Mutation, Inbreeding, Crossbreeding, and Selection in Evolution," *Proc. of 6th Int. Cong. Gen.*, Vol. 1, Ithaca, New York, 1932, pp. 356-366.

A Modified Lagrange's Interpolation for Image Reconstruction

Shan Suthaharan

*Department of Computer Science, Monash University,
Clayton, Melbourne, Australia 3168
e-mail:shan@bruce.cs.monash.edu.au*

Abstract

Interpolation techniques are being used in image restoration to construct a noise reduced image from a degraded image. In these techniques, the intensity of a pixel is obtained by interpolating the intensity values of the nearest surrounding pixels. The standard two-dimensional Lagrange's interpolation technique does not allow the use of all the eight nearest surrounding pixels and this is the limitation of this technique. In this paper, a modification is suggested to this technique so that it can use all the eight pixel intensity values in the interpolation. Interpolation techniques reduce the noise while introducing an error called interpolation error. The modified Lagrange's interpolation technique suggested in this paper reduces both the noise and the interpolation error, but comparable to the other interpolation techniques such as simple averaging and weighted averaging of the surrounding pixels, the noise is increased and it is shown that the increase is insignificant.

Keywords: simple averaging, weighted averaging, Lagrange's interpolation, interpolation error, noise reduction, first degraded image, second degraded image.

1. Introduction

In this paper, a method is proposed to construct a noise reduced image from an existing degraded image by modified standard two-dimensional Lagrange's interpolation technique. The existing and the constructed degraded images are referred to as the *first* and the *second* degraded images. It is assumed that the first degraded image s_1 satisfies the following linear spatially-invariant degradation model [1,3]:

$$s_1[i, j] = p[i, j] \otimes\otimes f[i, j] + n_1[i, j], \quad (1)$$

where the first term $p[i, j] \otimes\otimes f[i, j]$ on the right hand side of this imaging equation is called *noise-free* image and the noise $n_1[i, j]$ with mean zero is uncorrelated to the original image $f[i, j]$ and $(i, j) = (1, 1) \dots (M \times N)$.

In this proposed method, the intensity value $s_2[i, j]$ of the $[i, j]$ th pixel of the second degraded image s_2 is calculated using the nearest eight neighbouring pixels of the $[i, j]$ th pixel of the first degraded image s_1 . The intensity values of these neighbouring pixels are shown in Figure 1. In this figure, the centre pixel is denoted by $s_2[i, j]$ to indicate that it will be considered as the intensity value of the second degraded image s_2 at the $[i, j]$ th pixel. This intensity value $s_2[i, j]$ is calculated by averaging the intensity values of the eight neighbouring pixels with appropriate weightings. These weightings are represented by $k[c, d]$ and are shown in Figure 2, where $c, d = -1, 0, +1$. This is called the *interpolation mask* and the second degraded image is constructed using this mask over the whole of the first degraded image.

The weightings $k[c, d]$ values are selected in such a way that the four corner pixels contribute equally to the centre

pixel, while the remaining four pixels also contribute equally, but different (or equal) to the corner pixels as shown in Figure 3(a). Based on this assumption, three interpolation algorithms are suggested to choose the appropriate $k[c, d]$ values. These algorithms are:

1. **Simple averaging:** In this algorithm, discussed in detail in section 3.1, the weightings of the interpolation mask are assigned an equal value.
2. **Weighted averaging:** In this algorithm, the weightings are selected in such a way that they are inversely proportional to their distances from the centre pixel. Detailed discussion is presented in section 3.2.
3. **Modified Lagrange's interpolation:** In this algorithm, discussed in detail in section 3.3, the weightings are calculated using the modified Lagrange's interpolation.

The standard two-dimensional Lagrange's interpolation technique cannot be applied in image restoration to construct the second degraded image, because it does not allow the use of all the eight nearest neighbouring pixels in the interpolation [2]. The modified Lagrange's interpolation proposed in this paper makes use of all the eight neighbouring pixels.

2. The Interpolation

In this section, the second degraded image s_2 is constructed by calculating the intensity value $s_2[i, j]$ as the interpolation of the nearest eight neighbouring pixels of $s_1[i, j]$ of the first degraded image s_1 . This interpolation is defined by the following expression:

$$s_2[i, j] = \frac{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] s_1[i+c, j+d]}{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d]}, \quad (2)$$

where $(i, j) = (1, 1) \dots (M \times N)$ and $k[c, d]$ represents the weighting of the pixel $[i+c, j+d]$ of the first degraded image and the value of $k[0, 0]$ is zero. The $k[0, 0]$ is assigned a value zero so that the intensity value $s_2[i, j]$ of the second degraded image is independent of the intensity value $s_1[i, j]$ of the first degraded image.

$s_1[i-1, j-1]$	$s_1[i-1, j]$	$s_1[i-1, j+1]$
$s_1[i, j-1]$	$s_2[i, j]$	$s_1[i, j+1]$
$s_1[i+1, j-1]$	$s_1[i+1, j]$	$s_1[i+1, j+1]$

Figure. 1: Eight neighbouring pixels: The intensity values of the nearest eight neighbouring pixels of the $[i, j]$ th pixel of the first degraded image and the intensity value of the $[i, j]$ th pixel of the second degraded image.

$k[-1, -1]$	$k[-1, 0]$	$k[-1, +1]$
$k[0, -1]$	$k[0, 0]$	$k[0, +1]$
$k[+1, -1]$	$k[+1, 0]$	$k[+1, +1]$

Figure. 2: The interpolation mask: The 3x3 mask used as the weights in the proposed interpolation algorithms. The value of the centre weight $k[0, 0]$ is zero.

Using the interpolation expression in equation (2) over the whole of the first degraded image, the second degraded image can be constructed. However, when the boundary pixel intensity values of the second degraded image are calculated, not all of the eight neighbouring pixels of the first degraded image are available. Therefore, this boundary value problem must be resolved before applying the interpolation. This can be achieved

by extending the rows and columns at the boundary with their same values respectively. This is discussed below.

A Solution to the Boundary Value Problem

The boundary pixel values of the second image, such as $s_2[i, 1]$, $s_2[i, N]$, $s_2[1, j]$ and $s_2[M, j]$ cannot be calculated, because the neighbouring pixels, such as $s_1[i, 0]$, $s_1[i, N+1]$, $s_1[0, j]$ and $s_1[M+1, j]$, of $s_1[i, j]$ are not available. Therefore the following assumptions are made:

$$\begin{aligned} s_1[i, 0] &= s_1[i, 1], & s_1[i, N+1] &= s_1[i, N], \\ s_1[0, j] &= s_1[1, j], & s_1[M+1, j] &= s_1[M, j], \end{aligned} \quad (3)$$

where $i = 1 \dots M$ and $j = 1 \dots N$,

so that all the eight neighbouring pixel intensity values are available for the calculation of the values of the boundary pixels of the second degraded image.

2.1 Selecting $k[c, d]$ values

In this section, a method is suggested to obtain the $k[c, d]$ values so that the second degraded image is constructed using the equation (2). In this method the $k[c, d]$ values are selected such that

$$\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] = 1. \quad (4)$$

This normalization is done so that the interpolation method does not introduce a brightness bias in the second degraded image. It is also assumed that all four corner pixels contribute equally to the centre pixel and the remaining four pixels also contribute equally, but different in a manner to the corner pixels. This assumption is shown in Figure 3(a), where the contribution of the corner pixels is weighted as a and the contribution of the remaining pixels is weighted as b . Using the equation (4) and the weightings a and b , we can write,

$$4a + 4b = 1. \quad (5)$$

That is

$$a + b = 0.25, \quad (6)$$

where a and b can assume a value between 0 and 0.25.

From this equation, we can see that the minimum and maximum values of a and b are 0 and 0.25 respectively, and a is minimum when b is maximum and vice versa. In other words, when a is minimum, only the nearest four pixels are used in the interpolation and when b is

minimum, only the corner pixels are used.

2.2 Degradation Model for the Constructed Image

In this section, it is proved that the second degraded image constructed by the interpolation equation in (2) satisfies the linear spatially-invariant degradation model. It is also shown that the point spread function of the first degraded image can be used as the point spread function of the second degraded image. The error caused by this approximation is discussed in this section. The proofs of these results are given in the following section.

2.2.1 Derivation of the Model

Let us represent the numerator of the equation (2) by $g[i, j]$. That is:

$$g[i, j] = \sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] \cdot s_1[i + c, j + d]. \quad (7)$$

In this equation, the term $s_1[i + c, j + d]$ is the intensity value of the first degraded image at pixel $[i + c, j + d]$ and this can be written, using the equation (1), as follows:

$$\begin{aligned} s_1[i + c, j + d] &= p[i + c, j + d] \otimes\otimes f[i + c, j + d] \\ &+ n_1[i + c, j + d]. \end{aligned} \quad (8)$$

Substituting this equation into the equation (7), we can obtain the following relationship:

$$\begin{aligned} g[i, j] &= \\ &\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] (p[i + c, j + d] \otimes\otimes f[i + c, j + d] \\ &+ n_1[i + c, j + d]). \end{aligned} \quad (9)$$

By expanding the expression on the right hand side of the above equation, we get the following:

$$\begin{aligned} g[i, j] &= \\ &\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] (p[i + c, j + d] \otimes\otimes f[i + c, j + d]) \\ &+ \sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] n_1[i + c, j + d]. \end{aligned} \quad (10)$$

Hence, by dividing both sides of the equation (10) by $\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d]$, we have

$$\frac{g[i, j]}{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d]} =$$

$$\begin{aligned} &\frac{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] (p[i + c, j + d] \otimes\otimes f[i + c, j + d])}{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d]} \\ &+ \frac{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] n_1[i + c, j + d]}{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d]}. \end{aligned} \quad (11)$$

The first expression on the right hand side of this equation is the weighted average of the nearest eight neighbouring pixels of the $[i, j]$ th pixel of the noise-free image. This expression can be approximated to the $[i, j]$ th pixel intensity value of the noise-free image, $p[i, j] \otimes\otimes f[i, j]$. That is:

$$\begin{aligned} p[i, j] \otimes\otimes f[i, j] &= \\ &\frac{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] (p[i + c, j + d] \otimes\otimes f[i + c, j + d])}{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d]} \end{aligned} \quad (12)$$

The error introduced by this approximation is called *interpolation error*. The lower the interpolation error, is the greater the accuracy of the model. This interpolation error will be amplified with the larger mask and therefore the smallest mask size of 3×3 is selected.

Using this approximation, the equation (11) can also be written as follows:

$$\begin{aligned} \frac{g[i, j]}{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d]} &= p[i, j] \otimes\otimes f[i, j] \\ &+ \frac{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] n_1[i + c, j + d]}{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d]}. \end{aligned} \quad (13)$$

Thus, this equation can be considered as the second degraded image which satisfies the following linear spatially-invariant degradation model:

$$s_2[i, j] = p[i, j] \otimes\otimes f[i, j] + n_2[i, j], \quad (14)$$

where the intensity value $s_2[i, j]$ (using equations (7) and (13)) is:

$$s_2[i, j] = \frac{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] s_1[i + c, j + d]}{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d]}. \quad (15)$$

and the noise $n_2[i, j]$ is:

$$n_2[i, j] = \frac{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] n_1[i + c, j + d]}{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d]} . \quad (16)$$

This shows that if the second degraded image was constructed by the interpolation equation given in (2), using the appropriate $k[c, d]$ values, then this image would approximately satisfy the same imaging equation as of the first degraded image with the same point spread function.

In a previous paper, a method was introduced to estimate Γ , which is the *a priori* representation of the signal-to-noise ratio, using two images known as the first and the second degraded images, where the second degraded image is constructed from the first degraded image such that the point spread function of the second degraded image is the same as of the first degraded image [6]. This assumption is true only if the interpolation error is negligible. Therefore, the $k[c, d]$ values should be selected so as to reduce the interpolation error.

2.2.2 Analysis of the Interpolation Error

In this section, the interpolation errors with and without noise are discussed and experimented. These errors are denoted by E_{pf} and E_{pf_n} respectively and defined by

$$E_{pf} = p[i, j] \otimes \otimes f[i, j] - \frac{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] (p[i + c, j + d] \otimes \otimes f[i + c, j + d])}{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d]} \quad (17a)$$

$$E_{pf_n} = E_{pf} \frac{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] n_1[i + c, j + d]}{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d]} . \quad (17b)$$

In the experiment, the error E_{pf} and E_{pf_n} are calculated for the second degraded images of *human face*, *mug*, *saturn*, *discs*, *bubbles* and *rocks* in Figures 4(a) through 4(f).

In the following section, both the images that are degraded only by the point spread function and that are degraded both by the point spread function and the noise are considered. To measure the interpolation error in equation (17a) for different values of weighting a , we should interpolate the images that are only degraded by the point spread function. If the images that are degraded by both point spread function and noise are interpolated, then the resultant error is the combination of the interpolation error and the noise, where the interpolation

error cannot be identified (or separated from the noise).

a	human face	mug	saturn	discs	bubbles	rocks
0.00	0.46	0.38	0.21	0.16	0.92	19.95
0.01	0.49	0.40	0.22	0.17	0.98	21.42
0.02	0.51	0.42	0.23	0.18	1.05	22.94
0.03	0.54	0.44	0.25	0.19	1.12	24.51
0.04	0.58	0.47	0.26	0.21	1.20	26.13
0.05	0.61	0.49	0.27	0.22	1.27	27.81
0.06	0.64	0.52	0.29	0.24	1.35	29.54
0.07	0.68	0.55	0.30	0.25	1.43	31.32
0.08	0.71	0.57	0.32	0.27	1.52	33.15
0.09	0.75	0.60	0.33	0.28	1.61	35.04
0.10	0.79	0.63	0.35	0.30	1.70	36.98
0.11	0.83	0.67	0.36	0.31	1.79	38.98
0.12	0.87	0.70	0.38	0.33	1.88	41.02
0.13	0.92	0.73	0.40	0.35	1.98	43.13
0.14	0.96	0.77	0.42	0.37	2.08	45.28
0.15	1.00	0.80	0.44	0.38	2.18	47.49
0.16	1.05	0.84	0.46	0.40	2.29	49.74
0.17	1.10	0.87	0.48	0.42	2.40	52.06
0.18	1.15	0.91	0.50	0.44	2.51	54.42
0.19	1.20	0.95	0.52	0.46	2.62	56.84
0.20	1.25	0.99	0.54	0.48	2.74	59.31
0.21	1.30	1.03	0.56	0.50	2.86	61.84
0.22	1.36	1.08	0.58	0.53	2.98	64.41
0.23	1.41	1.12	0.61	0.55	3.11	67.05
0.24	1.47	1.16	0.63	0.57	3.23	69.73
0.25	1.53	1.21	0.65	0.59	3.36	72.47

Table I: Interpolation errors without noise. The images of *human face*, *mug*, *saturn*, *discs* and *bubbles* are blurred by the Gaussian point spread function with variance 7 and the image of *rocks* is blurred by the Gaussian point spread function with variance 2 and interpolated for different values of a within the range 0 and 0.25.

Interpolation Error without noise

In this section, the interpolation error (without noise) introduced by the different values of a is considered. In order to carry out this task, the original images in Figures 4(a) through 4(e) are blurred by the Gaussian point spread function with variance 7 and the image in Figure 4(f) is interpolated by the gaussian point spread function with variance 2. These blurred images are interpolated using the equation (2) with different values of a to construct the second images. The mean square errors between the constructed images and their original images are given in Table I. From this table, we can conclude that the interpolation error is minimum for all the images when a is zero.

The significance of these values depends on the errors caused by the point spread function on the images. Therefore, the mean square errors between the blurred and the original images are calculated and are given in Table II. Comparing the values in Tables I and II, we can

conclude that the interpolation error values are negligible.

human face	mug	saturn	discs	bubbles	rocks
141.55	156.91	134.67	248.85	479.81	2808.90

Table II: Mean square errors between the blurred and the original images.

a	human face	mug	saturn	discs	bubbles	rocks
0.00	60.64	68.89	241.58	363.35	152.69	177.84
0.01	55.08	63.61	220.36	339.01	138.89	162.37
0.02	49.88	58.57	199.99	315.34	130.31	155.46
0.03	45.05	53.78	180.50	292.35	124.81	149.60
0.04	40.58	49.25	161.92	270.09	119.53	144.10
0.05	36.72	44.98	151.69	248.56	114.47	140.58
0.06	36.12	40.98	144.09	227.81	109.63	136.37
0.07	35.68	37.26	136.74	223.38	104.25	132.35
0.08	35.38	33.83	129.58	222.39	97.88	128.52
0.09	35.22	30.70	122.61	221.55	91.90	124.88
0.10	35.22	27.95	115.93	220.40	86.32	121.44
0.11	35.36	25.55	109.55	219.11	81.70	118.20
0.12	35.65	23.56	104.12	217.91	79.15	114.90
0.13	36.09	24.16	104.26	216.84	76.85	110.54
0.14	36.42	26.28	110.78	215.91	78.11	106.48
0.15	36.71	28.90	119.35	215.11	83.79	108.72
0.16	37.19	31.85	128.34	214.45	89.79	115.84
0.17	37.85	35.13	137.40	213.93	96.10	123.32
0.18	38.70	38.71	146.85	220.57	102.71	131.16
0.19	39.73	42.59	156.66	233.50	109.63	139.35
0.20	40.95	46.77	166.85	246.26	116.83	147.87
0.21	42.34	51.24	177.32	259.18	124.32	156.73
0.22	45.71	55.98	187.98	272.41	132.09	165.92
0.23	50.90	61.01	201.50	285.95	143.61	180.35
0.24	56.47	66.30	218.06	299.77	159.82	199.93
0.25	62.39	71.84	235.20	318.01	176.85	220.43

Table III: Interpolation Errors with noise. The blurred images used in Table I are corrupted by the Gaussian noise with mean zero and variance 100.

Interpolation Error with noise

In this case, the blurred images obtained in the previous section are corrupted by an additive Gaussian noise with mean zero and variance 100, where the noise is uncorrelated to the original images. The second degraded images are constructed by interpolation using the equation (2) and then the mean square errors between the constructed images and their blurred images are calculated and are shown in Table II. These mean square errors are the result of both the interpolation error and

the noise. From Table III, we can conclude that the error is minimum for human face, mug, saturn, discs and rocks at $a = 0.1$, $a = 0.12$, $a = 0.12$, $a = 0.17$, $a = 0.13$ and $a = 0.12$ respectively.

Based on this analysis, we can see from Table I that the interpolation error (without noise) is minimum at $a = 0$ and the interpolation error with noise is minimum (on average) at a value 0.12. This shows that the simple averaging is the most effective way to reduce the noise [5]. However, our aim is to select a such that the interpolation error is low while not increasing the noise significantly.

Therefore, we need an algorithm to select an appropriate value for a within the range 0 to 0.12. This follows the fact that the contribution from the nearest four pixels is higher than the corner pixels. In other words, a higher weighting must be assigned to the nearest pixel than the corner pixels. Determining the ideal value for this weighting is critical to minimize the interpolation error while not increasing the noise significantly. To satisfy this condition, three interpolation algorithms are analyzed in the following discussions.

2.2.3 Analysis of Noise Reduction

Now let us look at the relationship between the noise present in the first and the second degraded images. By taking the statistical variance on both sides of the equation (16) and using the assumption that $n_1[i, j]$ is identically and independently distributed, the following result is obtained:

$$\text{Var}(n_2[i, j]) = w^2 \cdot \text{Var}(n_1[i, j]), \quad (18)$$

where w is a constant and given by the following function of $k[c, d]$ values.:

$$w^2 = \frac{\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k^2[c, d]}{\left(\sum_{c=-1}^{+1} \sum_{d=-1}^{+1} k[c, d] \right)^2}. \quad (19)$$

This shows that if the $k[c, d]$ values are calculated by the interpolation, then the statistical variance of the noise present in the second degraded image can be calculated from the statistical variance of the noise present in the first degraded image. Also, since the expression on the denominator of the equation (19) is greater than the expression on the numerator, hence w^2 is always less than one. Therefore, from the equation (18), we can say that the noise in the second degraded image is always less than the noise in the first degraded image.

a	b	a	0.125	0.125	0.125
b	0	b	0.125	0	0.125
a	b	a	0.125	0.125	0.125

(a)

(b)

0.1035	0.1465	0.1035	0.05	0.20	0.05
0.1465	0	0.1465	0.20	0	0.20
0.1035	0.1465	0.1035	0.05	0.20	0.05

(c)

(d)

Figure 3:a) Generalize weights for the interpolation algorithms. b) Weights used for simple averaging algorithm. c) Weights used for weighted averaging algorithm. d) Weights obtained by the modified Lagrange's interpolation algorithm.

3. Interpolation Algorithms

The simple averaging, weighted averaging and modified Lagrange's interpolation algorithms are discussed in subsections 3.1, 3.2 and 3.3 respectively.

3.1 Simple Averaging

In this algorithm, the intensity value $s_2[i, j]$ of the second degraded image is computed by taking the simple average of its eight neighbouring pixels. Figure. 3(b) shows the weighting $k[c, d]$ which equals 0.125 and is used in this algorithm. If we substitute these $k[c, d]$ values in the equation (2), we get the following:

$$s_2[i, j] = \sum_{c=-1}^{+1} \sum_{d=-1}^{+1} 0.125 \cdot s_1[i + c, j + d]. \quad (20)$$

The simple averaging algorithm assumes that the four corner pixels and the other four closests pixels have the same contribution to the centre pixel.

3.2 Weighted Averaging

In this algorithm the weightings $k[c, d]$ are chosen in such a way that they are inversely proportional to their

distances from the $[i, j]$ th pixel. Figure 3(c) shows these weightings. If these $k[c, d]$ values are used in equation 2, it becomes:

$$s_2[i, j] = \sum_{c=-1}^{+1} \sum_{d=-1}^{+1} \{0.1035, 0.1465\} \cdot s_1[i + c, j + d], \quad (21)$$

where $\{0.1035, 0.1465\}$ represents that the weighting of 0.1035 is selected for the corner pixels and 0.1465 is selected for remaining pixels. This is based on the assumption that the contribution of a neighbouring pixel decreases with its distance from the pixel of interest.

3.3 Modified Lagrange's Interpolation

As a guide to the demonstration of the standard two-dimensional Lagrange's interpolation, let us consider the points (x_i, y_j, f_{ij}) shown in Figure 4, where f_{ij} is the actual function value at (x_i, y_j) and $i, j = -1, 0, 1$. An arbitrary point (x, y) is used to demonstrate this interpolation method in the two-dimensional Cartesian coordinate system.

(x_{-1}, y_{-1})	.	(x_{-1}, y_0)	(x_{-1}, y_1)
(x_0, y_{-1})	.	(x_0, y_0)	(x_0, y_1)
(x, y)			
(x_1, y_{-1})	.	(x_1, y_0)	(x_1, y_1)

Figure 4.

The Standard one-dimensional Lagrange's interpolation method is initially applied horizontally in the x-direction to all three rows in Figure 4. The corresponding interpolated coefficients are $P_{-1}(x)$, $P_0(x)$ and $P_1(x)$, where [4]:

$$\begin{aligned} P_{-1}(x) &= \frac{(x - x_0)(x - x_1)}{(x_{-1} - x_0)(x_{-1} - x_1)} \cdot f_{-1-1} + \frac{(x - x_{-1})(x - x_1)}{(x_0 - x_{-1})(x_0 - x_1)} \cdot f_{0-1} \\ &\quad + \frac{(x - x_0)(x - x_1)}{(x_1 - x_0)(x_1 - x_1)} \cdot f_{1-1} \end{aligned} \quad (22)$$

$$\begin{aligned} P_0(x) &= \frac{(x - x_0)(x - x_1)}{(x_{-1} - x_0)(x_{-1} - x_1)} \cdot f_{-10} + \frac{(x - x_{-1})(x - x_1)}{(x_0 - x_{-1})(x_0 - x_1)} \cdot f_{00} \\ &\quad + \frac{(x - x_{-1})(x - x_0)}{(x_1 - x_{-1})(x_1 - x_0)} \cdot f_{10} \end{aligned} \quad (23)$$

and

$$P_1(x) = \frac{(x - x_0)(x - x_1)}{(x_{-1} - x_0)(x_{-1} - x_1)} \cdot f_{-11} + \frac{(x - x_{-1})(x - x_1)}{(x_0 - x_{-1})(x_0 - x_1)} \cdot f_{01}$$

$$+ \frac{(x - x_{-1})(x - x_0)}{(x_1 - x_{-1})(x_1 - x_0)} \cdot f_{11} \quad (24)$$

Then, the interpolation method is applied vertically through the points that have the coefficients $P_{-1}(x)$, $P_0(x)$, $P_1(x)$. This yields,

$$\begin{aligned} P(x, y) = & \frac{(y - y_0)(y - y_1)}{(y_{-1} - y_0)(y_{-1} - y_1)} \cdot P_{-1}(x) + \frac{(y - y_{-1})(y - y_1)}{(y_0 - y_{-1})(y_0 - y_1)} \cdot P_0(x) \\ & + \frac{(y - y_{-1})(y - y_0)}{(y_1 - y_{-1})(y_1 - y_0)} \cdot P_1(x) \end{aligned} \quad (25)$$

This is called the standard two-dimensional Lagrange's interpolation. In general form it can be written as follows [2]:

$$P(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 L_i(x) \cdot L_j(y) \cdot f(x_i, y_j), \quad (26)$$

where

$$L_i(x) = \prod_{\substack{k=-1 \\ k \neq i}}^1 \frac{(x - x_k)}{(x_i - x_k)}, \quad L_j(y) = \prod_{\substack{k=-1 \\ k \neq j}}^1 \frac{(y - y_k)}{(y_j - y_k)}. \quad (27)$$

Now, let us suppose that the middle point (x_0, y_0) is not known and we need to interpolate the value using the other points and by using the standard Lagrange's interpolation. Using the same concept as before, we can obtain the interpolation value just by interpolating (x_0, y_{-1}) and (x_0, y_1) or (x_{-1}, y_0) and (x_1, y_0) or the corner points. This shows that not all four nearest points can be used in the interpolation. This is the limitation of the Lagrange's interpolation. The nearest four points make a significant contribution to the centre point and therefore the use of all these points in the interpolation is vital. In this section, the Lagrange's interpolation method is modified so that all the surrounding eight points are used in the interpolation.

Modification

In this modification, the coefficients L_i and L_j are combined as $L_{i,j}$ and calculated using the Euclidean distances of the surrounding points. This gives

$$P(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 L_{i,j}(x, y) f(x_i, y_j), \quad (28)$$

where

$$L_{i,j}(x, y) = \prod_{\substack{(k,l)=(i,j) \\ (k,l) \neq (-1,-1)}}^{(1,1)} \frac{\sqrt{(x - x_k)^2 + (y - y_l)^2}}{\sqrt{(x_i - x_k)^2 + (y_j - y_l)^2}}, \quad (29)$$

and $(i, j) \neq (0, 0)$, because the centre point (x_0, y_0) is interpolated and it is assumed the function value $f(x_0, y_0)$ is not available at (x_0, y_0) .

In order to test if $P(x, y)$ passes through all the eight

points we need to show that.

$$P(x_p, y_q) = f(x_p, y_q), \quad (30)$$

where $(p, q) = (-1, -1) \dots (1, 1)$ and $(p, q) \neq (0, 0)$.

Proof:

Let us first prove that $P(x, y)$ passes through the point (x_{-1}, y_{-1}) . Substituting this point for (x, y) in the equation (28), we get

$$P(x_{-1}, y_{-1}) = \sum_{i=-1}^1 \sum_{j=-1}^1 L_{i,j}(x_{-1}, y_{-1}) f(x_i, y_j), \quad (31)$$

This can also be written as:

$$\begin{aligned} P(x_{-1}, y_{-1}) = & L_{-1,-1}(x_{-1}, y_{-1}) \cdot f(x_{-1}, y_{-1}) \\ & + \sum_{i=-1}^1 \sum_{j=-1}^1 L_{i,j}(x_{-1}, y_{-1}) f(x_i, y_j), \end{aligned} \quad (32)$$

Let us first calculate the modified Lagrange's coefficient $L_{-1,-1}(x_{-1}, y_{-1})$ in equation (32). Substituting $i = -1$, $j = -1$, $x = x_{-1}$ and $y = y_{-1}$ in equation (29), we have the following:

$$L_{-1,-1}(x_{-1}, y_{-1}) = \prod_{\substack{(k,l)=(-1,-1) \\ (k,l) \neq (-1,-1)}}^{(1,1)} \frac{\sqrt{(x_{-1} - x_k)^2 + (y_{-1} - y_l)^2}}{\sqrt{(x_{-1} - x_k)^2 + (y_{-1} - y_l)^2}}, \quad (33)$$

The numerator and denominator of the expression on the right hand side of this equation are the same and thus

$$L_{-1,-1}(x_{-1}, y_{-1}) = 1. \quad (34)$$

Let us now calculate $L_{i,j}(x_{-1}, y_{-1})$ for $(i, j) \neq (-1, -1)$ in equation (32). Substituting $x = x_{-1}$ and $y = y_{-1}$ in equation (29), we get

$$L_{i,j}(x_{-1}, y_{-1}) = \prod_{\substack{(k,l)=(i,j) \\ (k,l) \neq (-1,-1)}}^{(1,1)} \frac{\sqrt{(x_{-1} - x_k)^2 + (y_{-1} - y_l)^2}}{\sqrt{(x_{-1} - x_k)^2 + (y_j - y_l)^2}}, \quad (35)$$

Using this expression we can calculate the modified Lagrange's coefficients $L_{-1,0}(x_{-1}, y_{-1})$, $L_{-1,1}(x_{-1}, y_{-1})$, $L_{0,-1}(x_{-1}, y_{-1})$, $L_{0,1}(x_{-1}, y_{-1})$, $L_{1,-1}(x_{-1}, y_{-1})$, $L_{1,0}(x_{-1}, y_{-1})$ and $L_{1,1}(x_{-1}, y_{-1})$ in the second expression of equation (32). For example, the coefficient $L_{-1,0}(x_{-1}, y_{-1})$ can be calculated as follows:

$$L_{-1,0}(x_{-1}, y_{-1}) = \prod_{\substack{(k,l)=(-1,0) \\ (k,l) \neq (-1,-1)}}^{(1,1)} \frac{\sqrt{(x_{-1} - x_k)^2 + (y_{-1} - y_l)^2}}{\sqrt{(x_{-1} - x_k)^2 + (y_0 - y_l)^2}}, \quad (36)$$

In the multiplication on the right hand side of this equation the index (k, l) can take $(-1, -1)$ and this makes the expression 0. That is

$$L_{-1,0}(x_{-1}, y_{-1}) = 0. \quad (37)$$

Similarly other coefficients can be proved to be zero. Substituting the results of equations (34) and (37) in equation (32), we get

$$P(x_{-1}, y_{-1}) = f(x_{-1}, y_{-1}) \quad (38)$$

Similarly we can prove,

$$\begin{aligned} P(x_{-1}, y_0) &= f(x_{-1}, y_0), \quad P(x_{-1}, y_1) = f(x_{-1}, y_1), \\ P(x_0, y_{-1}) &= f(x_0, y_{-1}), \quad P(x_0, y_1) = f(x_0, y_1), \\ P(x_1, y_{-1}) &= f(x_1, y_{-1}), \quad P(x_1, y_0) = f(x_1, y_0), \\ P(x_2, y_2) &= f(x_2, y_2). \end{aligned} \quad (39)$$

Thus, $P(x, y)$ passes through all the eight points. That is the modification makes use of all the eight points in the interpolation.

Because we are trying to interpolate the centre pixel (x_0, y_0) , we need to calculate $P(x_0, y_0)$. In order to do this, the coefficients $L_{i,j}(x_0, y_0)$, $i, j = -1, 0, 1$, in equation (28) need to be calculated first. Substituting $x = x_0$ and $y = y_0$ in equation (29) we get,

$$L_{i,j}(x_0, y_0) = \prod_{\substack{(k,l)=(-1,-1) \\ (k,l)\neq(i,j)}}^{(1,1)} \frac{\sqrt{(x_0 - x_k)^2 + (y_0 - y_l)^2}}{\sqrt{(x_i - x_k)^2 + (y_j - y_l)^2}}, \quad (40)$$

Let us now convert this modified expression to the second image construction process. This is achieved by substituting $(x_0, y_0) = (0, 0)$, $(x_{-1}, y_{-1}) = (-1, -1)$, $(x_{-1}, y_0) = (-1, 0)$, $(x_{-1}, y_1) = (-1, 1)$, $(x_0, y_{-1}) = (0, -1)$, $(x_0, y_1) = (0, 1)$, $(x_1, y_{-1}) = (1, -1)$, $(x_1, y_0) = (1, 0)$ and $(x_1, y_1) = (1, 1)$ in equation (40). That is

$$L_{i,j}(0, 0) = \prod_{\substack{(k,l)=(-1,-1) \\ (k,l)\neq(i,j)}}^{(+1,+1)} \frac{\sqrt{(-k)^2 + (-l)^2}}{\sqrt{(i - k)^2 + (j - l)^2}}, \quad (41)$$

where $(i, j) \neq (0, 0)$.

Let us denote $L_{i,j}(0, 0)$ by $k[i, j]$. The above equation can be rewritten as follows:

$$k[i, j] = \prod_{\substack{(k,l)=(-1,-1) \\ (k,l)\neq(i,j)}}^{(+1,+1)} \frac{\sqrt{(-k)^2 + (-l)^2}}{\sqrt{(i - k)^2 + (j - l)^2}}, \quad (42)$$

Replacing (i, j) by (c, d) and (k, l) by (p, q) , the above expression can also be written as follows:

$$k[c, d] = \prod_{\substack{(p,q)=(-1,-1) \\ (p,q)\neq(c,d)}}^{(+1,+1)} \frac{\sqrt{(-p)^2 + (-q)^2}}{\sqrt{(c - p)^2 + (d - q)^2}}, \quad (43)$$

Using this expression let us calculate the weightings $k[-1, -1]$ and $k[-1, 0]$. The weighting $k[-1, 1]$ will be

$$k[-1, 1] = \prod_{\substack{(p,q)=(-1,-1) \\ (p,q)\neq(-1,1)}}^{(+1,+1)} \frac{\sqrt{(p)^2 + (q)^2}}{\sqrt{(1 + p)^2 + (1 + q)^2}}, \quad (44)$$

where $(p, q) \neq (0, 0)$ and $(p, q) \neq (-1, 1)$.

Multiplying these values we get $k[-1, -1] = 0.05$. Similarly, the weighting $k[-1, 0]$ will be

$$k[-1, 0] = \prod_{\substack{(p,q)=(-1,-1) \\ (p,q)\neq(-1,0)}}^{(+1,+1)} \frac{\sqrt{(p)^2 + (q)^2}}{\sqrt{(1 + p)^2 + (q)^2}}, \quad (45)$$

where $(p, q) \neq (0, 0)$ and $(p, q) \neq (-1, 0)$.

This gives the weighting $k[-1, 0] = 0.20$.

Because the interpolation mask is symmetric the other corner pixels weightings are also 0.05. Similarly remaining pixels weightings are 0.20. These weightings are shown in Figure 3(d). If we substitute these weightings in the equation (2), we get

$$s_2[i, j] = \sum_{c=-1}^{+1} \sum_{d=-1}^{+1} \{0.05, 0.20\} \cdot s_1[i + c, j + d], \quad (46)$$

where $\{0.05, 0.20\}$ means that the weighting 0.05 is selected for the corner pixels and the weighting 0.20 is selected for remaining four pixels. This supports the evidence that the contribution of a neighbouring pixel decreases when its distance from the pixel of interest increases.

4. Analysis of the Interpolation Algorithms

In this section, the noise reduction properties and interpolation error of the interpolation algorithms are discussed. Taking the statistical variance on both sides of the equations (20), (21) and (43), and assuming that the first noise $n_1[i, j]$ are identically and independently distributed, we can write the following relationships for the noise variances:

For Simple averaging:

$$\text{Var}(n_2[i, j]) = 0.125 \cdot \text{Var}(n_1[i, j]) \quad (47)$$

For Weighted averaging:

$$\text{Var}(n_2[i, j]) = 0.129 \cdot \text{Var}(n_1[i, j]) \quad (48)$$

For Lagrange's interpolation:

$$\text{Var}(n_2[i, j]) = 0.17 \cdot \text{Var}(n_1[i, j]) \quad (49)$$

The above relationships show that the noise variances of the second degraded images are reduced by the factors of 0.125, 0.129 and 0.17, when the second images are constructed by the simple averaging algorithm, the weighted averaging algorithm and the modified Lagrange's interpolation algorithm. This proves that the simple averaging algorithm is the most effective way of reducing the noise. However, our task is to minimize the interpolation error rather than the noise. We need to select an interpolation algorithm that can minimize the interpolation error while reducing the noise contamination from the first degraded image. In the next section, the effect of the simple averaging algorithm, the weighted averaging algorithm and the modified Lagrange's interpolation algorithm on edge regions and exponential regions are considered.

4.1 Effect on Edge Regions

In this analysis, the original edge image matrix in Figure 6(a) is interpolated by using the simple averaging, the weighted averaging and the modified Lagrange's interpolation algorithms. The second image matrices are given in Figures 6(b), 6(c) and 6(d) respectively. The original edge image matrix is subtracted from these second image matrices to measure the interpolation errors. The mean and the variance of these errors are calculated. The mean values are zero and the variances are 175.83, 156.25 and 112.57 respectively. This shows that the modified Lagrange's interpolation gives less error than the other two algorithms. As we can see from the Figures 6(b), 6(c) and 6(d), the intensity changes over the edge are {137, 162, 200}, {135, 164, 200} and {130, 170, 200} respectively. This shows that the modified Lagrange's interpolation causes less blur along the edges than the other two algorithms.

4.2 Effect on Exponential Regions

In this section, a 16x16 exponential image matrix is constructed. The following exponential function is used:

$$z = e^{(p \cdot x + q \cdot y)}. \quad (50)$$

where the parameters p and q can be chosen so that the function z takes significant values between 0 and 255.

The image generated with p = 0.21 and q = 0.15 is given in Figure 6(e).

In this analysis, the original exponential image in Figure 6(e) is interpolated by the simple averaging, the weighted averaging and the modified Lagrange's interpolation algorithms to construct the second images. Then the original image is subtracted from these second images to

calculate the interpolation errors. The variance of these errors are calculated and they are 6.05, 5.38 and 3.92 respectively. This shows that the modified Lagrange's interpolation gives less error than the other two algorithms for an image with exponential regions.

5. Conclusion

In this paper, a method was introduced to modify the standard two-dimensional Lagrange's interpolation. This modified Lagrange's interpolation was compared with the other interpolation techniques, namely simple averaging and weighted averaging.

In this technique, construction of the second image was obtained by an interpolation mask size of 3x3 using eight surrounding pixels. The weighting values in the mask were calculated using an expression derived by taking into consideration that the corner pixels contribute equally to the centre pixel and the other remaining pixels also contribute equally. The weighting of a pixel was calculated as a ratio of the contribution by the other remaining seven pixels to the centre pixel.

The limitation of not using all the eight nearest surrounding pixels by the standard two-dimensional Lagrange's interpolation technique is resolved by modifying the Lagrange's coefficients using the Euclidean distances of the pixels.

This modified Lagrange's interpolation reduces the interpolation error without increasing the noise level significantly. A Reduction in this interpolation error is needed to construct the second degraded image which has the same point spread function as the first degraded image and to use restoration techniques such as the one published in a previous paper [6].

Using all eight pixels with equal weighting reduces the noise, but introduces the interpolation error. Whereas excluding the corner pixels minimizes this interpolation error. Reducing the interpolation error without increasing the noise significantly is vital for the approximation used in the second image construction. In the modified Lagrange's interpolation, the weighting for the corner pixel is small compare to that of the nearest four pixels and this reduces the interpolation error. Based on the above discussion, we can conclude that this technique is more suitable in the construction of the second image.

6. References

- Blackledge, J. M., *Quantitative Coherent Imaging*, Academic Press, 1989.

2. Carnahan, B., Luther, H. A., and Wilkes, J. O., *Applied Numerical Methods*, John Wiley & Sons, 1969.
 3. Gonzalez, R. C., and Wintz, P., *Digital Image Processing*, Addison-Wesley, 1987.
 4. Kreyszig, E., *Advanced Engineering Mathematics*, John Wiley and Sons, sixth edition, 1988.
 5. Lim, J. S., *Two-Dimensional Signal and Image Processing*, Prentice-Hall 1990.
 6. Suthaharan, S., and Ray, S., "Auxiliary image construction by Lagrange's interpolation for estimation of Γ in image restoration," in *Image and Video Processing*, Proc. SPIE 1903, 1993, pp 84-90.

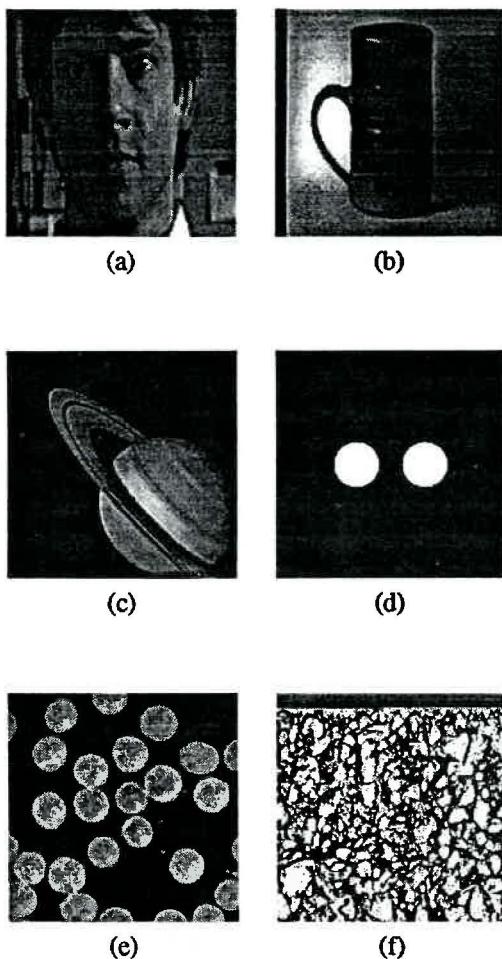


Figure 5. The original images of size 256 x 256. a) The original image of human face. b) The original image of mug. c) The original image of saturn. d) The original image of two discs, a synthetic image. e) The original image of bubbles. f) The original image of rocks.

(2)

1

6

1

1	1	1	1	1	2	2	2	3	3	4	5	6	7	8	9
1	1	1	1	2	2	3	3	4	4	5	5	6	7	8	10
1	1	2	2	2	3	3	4	5	5	6	7	9	10	12	14
1	2	2	2	3	3	4	5	6	7	8	9	11	13	15	17
2	2	3	3	4	4	5	6	7	8	10	12	14	16	18	21
2	3	3	4	5	6	7	8	9	11	12	14	17	20	23	27
3	4	4	5	6	7	8	10	11	13	15	18	21	24	28	33
4	5	5	6	7	9	10	12	14	16	19	22	26	30	35	41
5	6	7	8	9	11	13	15	17	20	24	27	32	37	43	50
6	7	8	10	12	14	16	18	21	25	29	34	40	46	54	62
8	9	11	12	14	17	20	23	27	31	36	42	49	57	66	77
10	11	13	15	18	21	24	28	33	38	45	52	60	70	82	95
12	14	16	19	22	26	30	35	41	47	55	64	75	87	101	121
15	17	20	24	27	32	37	43	50	59	68	79	92	107	125	145
18	21	25	29	34	40	46	54	62	72	84	98	114	132	154	179
23	27	31	36	42	49	57	66	77	90	104	121	141	164	190	221

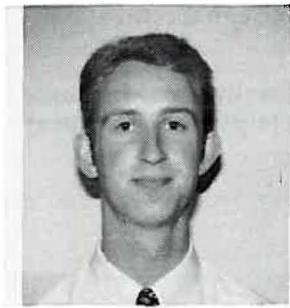
{e}

Figure 6. The image matrices of size 16×16 . a) An original edge image matrix. b) Image matrix (a) interpolated by the simple averaging ($a = 0.125$). c) Image matrix (a) interpolated by the weighted averaging ($a = 0.1035$). d) Image matrix (a) interpolated by the Lagrange's interpolation ($a = 0.05$). e) An original image matrix with an exponential shape.

Selected Author Biographies



George Cerkowicz - was born in Adelaide, South Australia, on March 18, 1972. He received a B.Eng in Electronic Engineering degree from the University of South Australia in 1993. In 1993 he joined AWA MicroElectronics Pty Ltd as a VLSI design engineer. His primary interests lies in the area of analog VLSI design whilst keeping a keen eye on the progress of artificial neural networks.



Malcolm R Haskard - is Professor of Microelectronics in the School of Electronic Engineering and founding Director of the Microelectronics Centre, both at the University of South Australia. He has a Masters Degree in Electronic Engineering, a Graduate Diploma in Mathematics and an Honours Degree in Electrical Engineering. A Fellow of the Institution of Engineers Australia, the Institution of Electrical Engineers (London). He is also a member of the International Society of Hybrid Microelectronics. Malcolm is the author of eight text books and of two invited chapters in other books. By invitation he has twice presented courses on behalf of UNESCO. His interests include, thick film hybrids and actuators and microengineering.

***Abstracts of the Australian and New Zealand Universities
Postgraduate Research Theses***

Degree: Doctor of Philosophy
Title: Automated ICEG Classification using Artificial Neural Networks
Author: Zherly Chi
Institution: Sydney University
Department: Systems Engineering and Design Automation Laboratory

This thesis presents artificial neural network techniques and architectures for intra-cardiac electrogram (ICEG) classification. The framework for the design of such classifiers is investigated and solutions are presented. The work has addressed ICEG classification in a "global" sense, that is, the requirements from the perspective of pre-processing and feature extraction, feature evaluation, classifier design, patient specific tuning, and postprocessing. As Implantable Cardioverter Defibrillators (ICDs) were a constant motivation, networks that are attractive from an implementation perspective are considered and architectural enhancements are provided.

The designed classifier has been evaluated on a large ICEG database that includes eight rhythms. In single chamber mode, a performance of 96% overall correct classification was achieved with 111 perfectly classified recordings out of a total of 143 recordings.

Degree: Doctor of Philosophy
Title: Arrhythmia Classification using Low Power VLSI
Author: Philip Leong
Institution: Sydney University
Department: Systems Engineering and Design Automation Laboratory

The implantable cardioverter-defibrillator (ICD) is a device which monitors the heart and delivers defibrillation and/or pacing therapy in the event of a life threatening arrhythmia. Such devices have had great success in preventing what otherwise would be sudden cardiac death. The arrhythmia classifier used in current devices makes its decision primarily based on rate criteria. Unfortunately, this method is very crude and can cause incorrect therapy to be delivered for certain arrhythmias, sometimes with fatal consequences.

This thesis describes a two chamber arrhythmia classification algorithm, called MATIC, which classifies heart signals based on timing and morphological criteria. MATIC attempts to mimic a cardiologist when making a classification and uses a decision tree to examine timing relations within and between channels. For most arrhythmias, this information is sufficient to make a reliable classification. For some arrhythmias, correct classification cannot be achieved based on rate alone, and in these cases, a neural network is used to analyse the morphology of the heart signal. The MATIC algorithms achieved 99.6% correct classification on a database of intracardiac electrogram signals containing 12483 QRS complexes recorded from 67 patients during electrophysiological studies.

The neural network used for morphology classification in the MATIC algorithm is computationally expensive and would have excessive power consumption for an implantable device. An analogue low power neural network VLSI chip called Kakadu was designed to address this problem. Kakadu's design is based on synapses implemented as multiplying digital to analogue converters which can operate at very low bias currents. Kakadu implements a three layer feedforward network with 10 inputs, 6 hidden units and 4 outputs and achieves typical power consumption figures of tens of microwatts. When Kakadu is incorporated in an arrhythmia classification system, power consumption of less than 25 nW can be achieved.

When Kakadu is used to provide the morphology classification of the MATIC classifier, the resulting system classifies arrhythmias both reliably and with low power consumption - the two necessary features of a classifier suitable for an ICD.

Abstracts of the Australian and New Zealand Universities Postgraduate Research Theses

Degree: Doctor of Philosophy
Title: General Macro Module Placement and Floorplanning
Author: Jun Wei Jiang
Institution: Sydney University
Department: Systems Engineering and Design Automation Laboratory

This dissertation focuses on VLSI macro module placement and floorplanning. Two efficient algorithm are presented: macro module placement under performance constraints; and floorplanning with optimised module area allocation and clock pin allocation.

The placement algorithm is based on an analytical approach. The problem and timing constraints are mathematically formulated into an objective function and related constraints. In the formulation, the objective function implicitly represents the total connection length; the nets on critical paths are assigned dynamic weights so that they are treated with different emphasis; routing areas are assigned by a dynamic estimator. To solve the formulated optimisation problem, mathematical calculations are mapped onto a set of physical moves, such as mirroring, shifting, re-shaping, etc. This results in improvements in the algorithm efficiency and the ability for 'hill climbing', which cannot be handled by traditional numerical programming algorithms. Better module orientations can be achieved because the corners of modules are treated independently.

The floorplanning algorithm consists of three phases: module adjacency topology formation, optimal module area allocation and clock pin allocation. Module adjacency topology formation is performed by a self-organisation technique which uses an artificial neural network. A constructive approach for optimally allocating module areas allows us to deal with both slicing and non-slicing adjacency topology. A novel property of module adjacency topology is used to measure the problem complexity and provide an upper limit on the search efficiency. For a given floorplan topology, the clock pin allocation problem is addressed by minimising the clock tree length while maintaining the zero skew of clock nets. The algorithms were implemented using the C language on a SUN SPARC-II and MCNC benchmarks used as tests. Very high quality results were obtained with a considerable improvement over previous works.

Degree: Doctor of Philosophy
Title: Nonlinear Analysis and Optimisation of SubCarrier Multiplexed Lightwave Systems
Author: Kamal E. Alameh
Institution: Sydney University
Department: Systems Engineering and Design Automation Laboratory

In this thesis, we present new accurate analysis, design and optimisation techniques for realising SubCarrier Multiplexed (SCM) lightwave video distribution systems that approach the fundamental limits of performance. The present work comprises original contributions which can be summarised as follows:

- * Design of ultra-low noise optical receiver for SCM systems, utilising for the first time the detailed effects of correlation between gate and drain FET noise sources to obtain partial noise cancellation in tuned optical receivers. An input tuning network circuit design is shown to be attractive in increasing the SCM distribution capacity.
- * Development of a novel, accurate theoretical analysis of nonlinear distortion in SCM lightwave transmitter, that determines the power spectral density of distortion noise. The analysis is general, in that arbitrary modulator transfer functions and arbitrary optical modulation index profiles can be handled.
- * Accurate prediction of the fundamental limits of transmission capacity of SCM lightwave systems.
- * Determination of the optimum optical modulation index for multi-channel SCM signal, under several modulation formats.
- * Elucidation for the first time, of the design factors involved in simultaneously optimising the laser overmodulation characteristics and the erbium fibre amplifier characteristics, for maximising the optical loss budget. An improved topology that incorporates an EDFA in conjunction with a linearised external modulation techniques is proposed and shown to be attractive for distribution capacity enhancement and video quality improvement.

Abstracts of the Australian and New Zealand Universities Postgraduate Research Theses

Degree: Doctor of Philosophy
Title: *Using Linda as the basis of an Operating System Microkernel*
Author: James Pinakis
Institution: The University of Western Australia
Department: Computer Science

Advances in computer hardware has led to the increased prevalence of systems comprised of many small processors connected by fast local area networks. The need to effectively manage the resources of these distributed systems has led to the development of distributed operating systems. The primary aim of a distributed operating systems is to hide the details of the computer and network hardware and provide the user with the view of a single, virtual machine. More recently, the feasibility of designing distributed operating systems using a microkernel has been demonstrated by systems such as Mach, Amoeba and CHORUS. A microkernel provides a minimal set of system services: a complete operating system is built on top of the microkernel as a collection of system servers.

This thesis investigates the use of Linda as the dominant model of process creation and communication in a microkernel-based operating system. Linda is a concurrent programming paradigm which permits cooperation between processes by controlling access to a shared data structure called a tuple space. The location-transparent communication and global namespace of Linda make it especially attractive for use in a distributed operating system kernel.

The microkernel proposed in this thesis provides a persistent Linda tuple space for use by both system and user level processes. Clients communicate via this tuple space using a set of language independent data types. The Linda paradigm is extended to support a new system data type, the ticket. Tickets allow processes to direct tuples for receipt by other processes, and also provide a mechanism by which processes can securely advertise their services in the tuple space.

The Linda paradigm is also extended to provide for the transfer of both heavyweight and lightweight processes through the tuple space. This allows more general process creation mechanisms to be employed: both types of processes can be remotely spawned by sending the binary to the appropriate server. The flexibility of this technique is illustrated by the implementation of Linda's eval operator as a user-level construct. In addition to providing a description of these extensions to the Linda paradigm, this thesis presents an implementation of a Linda system which has been used to verify these techniques. An overview of how a complete operating system can be designed using these constructs is provided, and some of the related difficulties are discussed. Finally, some examples of system servers are given.

Degree: Doctor of Philosophy
Title: *Practical Solution Caching for Prolog: An Explanation-Based Approach*
Author: Nick Lewins
Institution: The University of Western Australia
Department: Computer Science

Explanation-Based Learning (EBL) can be used to make inefficient Prolog programs more efficient. The improved efficiency is achieved by extracting search control knowledge from examples of program execution. This knowledge is used to guide the search for a solution, often in a more intelligent way than Prolog's default (depth-first) search strategy. This thesis investigates use of EBL to improve the efficiency of Prolog programs that are already efficient.

Prolog programmers write efficient programs. They craft programs so that the default search strategy will make appropriate search decisions. Such Prolog programs exhibit a low branching rate, and are therefore nearly deterministic. Such "near-deterministic" programs offer few opportunities for using search control knowledge. This thesis presents a method that augments a given near-deterministic Prolog program with a cache of fast special-case solution methods for frequently-occurring classes of top-level query. For each new top-level query, the cache entries are examined. If a special-case method in the cache suits the query, the cached method is used and the program is not invoked. This use of EBL is fundamentally different to that of previous systems. Whereas previous systems use EBL to guide search, this use of EBL does not. Rather, EBL is used to eliminate subroutine calls. Furthermore, EBL is used in a more restricted way than previous systems.

Abstracts of the Australian and New Zealand Universities Postgraduate Research Theses

Previous systems augment each predicate in the program with a cache rather than augmenting the entire program with a single cache.

Employing a single cache of solutions to top-level queries simplifies the reasoning required to ensure that the benefit of using the cache outweighs the cost of using it. A statistical method is employed to provide a probabilistic guarantee that each addition to the cache results in better overall performance. The ease of reasoning about the effect of cache entries also allows the system to dynamically adjust the generality of new entries. This adaptability allows performance improvements to be obtained for a wide range of programs, and in particular allows the system to be effective on recursive programs.

A set of experiments is presented that show the performance of the above approach over a set of Prolog programs. Time and space complexity analyses of the method are also presented. These results together indicate that the approach is indeed capable of optimising near-deterministic Prolog programs, and that its capabilities scale well with program complexity. It is also shown that the cost of improving such programs is reasonable, which implies that optimisation may be interleaved with problem-solving. The results suggest that the approach may be of practical benefit to Prolog programmers, by automatically adapting programs to their execution environment.

Degree: Doctor of Philosophy
Title: Depth Acquisition and Surface Reconstruction in three-dimensional Computer Vision
Author: Jesse Sheng Jin
Institution: University of Otago
Department: Department of Computer and Information Science

Vision is the crucial and dominant sensory system for gaining knowledge about the arrangement of objects and the occurrence of events in the environment. The idea of developing a computational theory of human vision is to explain the intricate processes underlying vision. Before doing so, we need a thorough understanding of how the human visual system performs. Observation and explanation go hand in hand, and it is in this way our stereo algorithms have been developed. We comment on the necessary, though conflicting, use of spatial and frequency information and develop a model utilising both. We also observe how linear interpolation occurs in human vision when other information is lacking, and develop a planar interpolation algorithm.

The use of Laplacian of Gaussian and Gabor filters is the most exciting event recently in the research of computational vision because they have a profile similar to the active profile of biological neurons. We use these filters to develop a multi-channel filtering scheme, which possesses the ability to extract abstract features, such as subjective contours, in a way similar to human perception.

We develop a novel approach using a Gaussian Sphere model to map the three dimensional space onto two stereoscopic views for depth acquisition in computational vision. It is too early at this stage to develop a comprehensive stereo visual system with all the facilities of human vision. However, this approach is the first to combine a monocular cue with binocular cues in stereo fusion as has been observed in the human visual system. We show how the mapping could be used to integrate multiple cues of depth perception in human vision into a unified frame.

Degree: Doctor of Philosophy
Title: The Temporal Transformational Data Model
Author: C. J. Robertson
Institution: University of Otago
Department: Department of Computer and Information Science

As a result of recent developments in temporal databases the user now has a choice between two types of history for an object. They can have either a snapshot history, where only the latest data values for the object are kept, or a full history, where all its data values since database creation are kept. These two history types provide for both ends of the spectrum of object histories but ignore all points in-between. All the values held

***Abstracts of the Australian and New Zealand Universities
Postgraduate Research Theses***

within a full history are assumed to have uniform importance (here called usefulness) to the user. This is not normally the case.

This thesis introduces the temporal transformational (T^2) data model. The new model fills the gap in the spectrum of temporal data models by providing the user with a facility for modelling usefulness. It allows the user to divide an object's history into different stages to reflect the changes in usefulness as the data ages. These stages are reflected in the object's storage and querying facilities. Through this approach, the ability to maintain shortened or partial histories of objects is provided. A new terminology is defined so that different aspects and stages of the history of an object, that have been previously unspecified, can be identified.

Initially the T^2 model is defined as an independent data model. Following this it is presented in terms of an extended temporal relational data model along with an outline of its implementation.

Degree: *Master of Science*
Title: *Constraint-Based Menu Extensions to Command Interpreters*
Author: *Jae Hoon Kim*
Institution: *The University of Western Australia*
Department: *Department of Computer Science*

Developing a quality user interface has long been an extremely difficult task, despite the wide availability of highly sophisticated computer graphics and advanced I/O technologies. One significant problem to be addressed is the manner in which the diversity of users' preferences for particular types of interface may be supported. This has often depended on their level of experience and their expectations. One promising approach in solving this problem has been to provide separate types of user interfaces or development tools for different classes of users. This approach, however, has failed to address the issues regarding the users' learning experience being inherently of a progressively growing nature, and that the user interface tools should be designed to consistently reflect this within a single uniform framework.

This thesis explores the prospect of unifying two of the most commonly used types of user interfaces, the menu-driven user interface and the traditional command line interpreter, in order to maximise the level of efficiency in using the resulting interface by different classes of users in one unified environment. This idea is demonstrated by developing a menu-driven command interpreter, Mshell, which effectively combines the command line oriented interactions and the menu oriented interactions without compromising either of their desirable characteristics. Mshell is implemented under the UNIX operating system, and utilises the X windowing environment.

In the Mshell environment, each command may be uniformly represented as menus which are designed and prototyped to create relevant contextual help which is immediately available. Menus are regarded as command line objects describing the operational subspace of each command, and they are specified and manipulated through command line interactions. The use of constraints in the Mshell specification language makes it possible to represent in the menu more tightly controlled semantics imposed by the construct of the command line arguments.

Degree: *Master of Engineering*
Title: *Generalization of Object-Oriented Components for Software Reuse*
Author: *Yagna Raj Pant*
Institution: *The University of New South Wales*
Department: *School of Computer Science and Engineering*

This thesis investigates the effort of generalizing object-oriented components for software reuse. The change in size and complexity of the components as a result of generalization is investigated.

Generalization and software reuse play important roles in the object-oriented paradigm as they have the potential for reducing the software development costs and improving the quality of the software developed. Reusable components provide benefits for future developments, yet their development requires some additional effort (long-term investment) in which potentially reusable components are generalized from existing, specific ones. However, in a "project culture" (where results, short-term profits, goals and final

Abstracts of the Australian and New Zealand Universities Postgraduate Research Theses

products are more important than the generalized components) it may be difficult to get management support for generalization and software reuse. In this thesis, a reuse model is proposed that may be more attractive to some organizations.

There are only a few empirical studies related to the effort required to generalize object-oriented classes for software reuse. This dearth led to the design and implementation of a pilot project based on C++ and measurement of effort both before and after the generalization process.

There are no widely-accepted size and complexity metrics for the object-oriented paradigm, although many do exist for the procedural environment. In this thesis, I have applied existing size and complexity metrics to the object-oriented paradigm. It is found that the use of inheritance and aggregation relationships within the object paradigm makes the notion of program size difficult to conceptualize and measure in terms of traditional metrics. I have also proposed a size and complexity measure (a simple, yet powerful extension of the normal lines of code) which combines size with the complexity of the programs. The proposed measure has the potential for use in measuring the effort related to understanding software for software maintenance, debugging and generalization.

Degree: Doctor of Philosophy
Title: *Distributed Real-Time Flight Simulation: Collisions, Terrain and Networks*
Author: Mark Andrew Bickerstaff
Institution: The University of New South Wales
Department: School of Computer Science and Engineering

This thesis makes contributions to three areas in the field of real-time flight simulation. The first contribution is a novel distributed and parallel architecture for managing real-time collision detection for an arbitrarily large collection of arbitrarily complex objects. Previously systems have attempted to solve to the complete $O(N^2)$ problem and become bound by available computational resources. The architecture proposed here partitions the problem amongst N workstations on a distributed real-time visual simulation system, and reduces the problem that must be solved per workstation to a complexity of $O(N)$. Within each workstation a parallel processing system performs a one-to-many polygon-polygon comparison without imposing any restriction on object complexity. An integrated circuit was designed and worked successfully at a speed very close to specification, in first silicon.

The second contribution is a novel network reconfiguration protocol and a proposal for a network architecture that caters for multiple faults in a distributed real-time visual simulation system. The protocol allows real-time recovery from faults without degrading the realism of the simulation. The network and the reconfiguration protocol are described in a hardware description language and are able to recover from cable faults within a maximum of ten video frame times.

The third contribution describes a new type of terrain database system which is potentially able to significantly reduce the storage required for large terrain databases, while maintaining a high level of surface detail without overloading the rendering system. The speculative database system consists of a method for terrain surface compression, a real-time terrain surface decompression system, a real-time global culling system for very large databases of three dimensional objects, and a system for level of detail control (based upon an adaptation of the collision detection hardware).

Degree: Doctor of Philosophy
Title: *Inductive Learning of Search Control Rules for Planning*
Author: C. Leckie
Institution: Monash University
Department: Department of Computer Science

In this thesis, we present a domain-independent inductive approach to learning search control knowledge for plan generation, and consider the advantages of this approach from the point of view of the utility problem. In addition, this research incorporates a practical method for estimating the overall utility of the derived search control rules in combination.

***Abstracts of the Australian and New Zealand Universities
Postgraduate Research Theses***

Approaches based on both explanation-based learning and inductive learning have previously been used to learn search control rules. Unlike explanation-based learning techniques, the approach presented in this thesis requires neither a domain theory nor domain-specific axioms or heuristics to simplify the descriptions that it learns. Further, the effectiveness of this approach is superior to that of previous inductive techniques from the perspective of the utility problem.

Our model for learning search control rules has been implemented in a system called Grasshopper, which learns about goal, operator and binding decisions in planning. Grasshopper analyses the search trees that are generated by a planning system, and suggests modifications to the planner's search strategy that will improve the efficiency of its search in the future. Given a number of training instances, Grasshopper looks for sets of similar decisions in order to form the consequents of new search control rules. Grasshopper then generates antecedents for these rules by characterising the problem solving contexts where these decisions occurred. Finally, Grasshopper looks for the optimum set of rules, where the rules have the maximum combined utility. This set is then passed to the planner for use on subsequent problems.

In this thesis, we describe Grasshopper and present an empirical evaluation of its performance on four different test domains using a standard planning system. This evaluation demonstrates the effectiveness of the rules learned by Grasshopper at reducing the total search time required for planning in each of the four test domains. In addition, we present an empirical comparison of the performance of our approach and that of a representative explanation-based learning system. In this comparison, the rules learned by Grasshopper produced greater reductions in search time than those rules learned by the explanation-based learning technique for all of the four domains tested. This comparison demonstrates that it is possible to learn effective search control rules without having to refer to a domain theory. Furthermore, it demonstrates that by learning from more than one example at a time, our inductive approach learns individual rules of greater utility than those learned by an explanation-based learning technique.

Degree:	<i>Doctor of Philosophy</i>
Title:	<i>A Probabilistic approach to plan recognition and response generation during cooperative consultations</i>
Author:	<i>Bhavani Laxman Raskutti</i>
Institution:	<i>Monash University</i>
Department:	<i>Department of Computer Science</i>

This thesis presents an integrated approach to cope with the uncertainty that is inherent in natural language communications. Uncertainty causes difficulties in recognizing a speaker's intentions from his/her utterances. Yet, people handle the uncertainty with extreme ease. This thesis is concerned with designing computerized information providers that emulate people's ability to handle uncertainty by first recognizing a user's plans from his/her queries, and then responding appropriately to these queries.

Uncertainty in understanding dialogues arises due to a number of factors. One of these factors is that there are different ways of relating a new statement to the previous discourse. Another factor contributing to the uncertainty is that different information sources have different reliability. The mechanism described in this thesis uses the Bayesian theory of probability to combine the contributions of the different factors to the overall probability of an inferred intention. It has been implemented in an information providing system for the travel domain called RADAR.

RADAR accounts for the uncertainty in natural language interactions by taking into consideration the characteristics of structure, coherence and relevance exhibited by information-seeking dialogues. The fact that dialogues have an expected structure is used in the architecture of RADAR and in the approach taken for performing inferences. The coherence of dialogues is used to assign probabilities to the interpretations generated from the discourse when assimilating new statements into the understanding gleaned from the previous discourse. The relevance of the presented information is measured by the information content of an interpretation. This measure also incorporates the reliability of the information sources of different inferences. The information content measure is used during the inference process both for preferring interpretations with more relevant information and as an informative stopping rule to terminate the inference process when an interpretation has a sufficient information content.

Abstracts of the Australian and New Zealand Universities Postgraduate Research Theses

The inferences performed by the information provider are analyzed and on the basis of the analysis, the computerized information provider performs one of the following actions: (1) formulate a plan to achieve an inferred goal, (2) generate a query to obtain additional information regarding a user's intentions, or (3) perform additional inferences to recognize the user's intentions. The generation of queries is based on the information content and the relative probabilities of the interpretations.

Degree: Doctor of Philosophy
Title: *Fast three-dimensional rendering using isoluminance contours*
Author: D. Conway
Institution: Monash University
Department: Department of Computer Science

"Isoluminance contouring" is a contour-based solids representation scheme which supports a set of techniques for fast, yet realistic, synthetic scene rendering. Objects are modelled as collections of contours which map contiguous points of equal brightness and chromaticity on their surface.

The construction of isoluminance contours for six simple classes of object (spheres, ellipsoids, cylinders, cones, truncii and polyhedra) is described. These objects are then used as the basis of a constructive solid geometry system, by introducing three regularized Boolean operators over contour sets (union, intersection and difference). A technique for computing isoluminance contours for an arbitrary implicit surface is also detailed.

A fast rendering algorithm for isoluminance contoured objects is presented. The algorithm caters for both primitives and composite objects and compares very favourably in terms of speed and final image quality with established rendering techniques.

The adaptation of various shading models for use with isoluminance contouring is discussed. Extensions to the basic technique are presented which permit the use of multiple illuminants and surfaces with composite (both specular and diffuse) characteristics.

The application of 2D and 3D patterns and 3D textures to isoluminance contoured objects is described. Techniques permitting shadowing, area and volume light sources, finite depth-of-field, motion blur and antialiasing are explained.

An extensive review of existing techniques for representation, shading and rendering of objects is also provided, in order to compare and contrast them with isoluminance contouring.

A series of empirical studies of the efficiency of the technique is presented and conclusions regarding the performance and applicability of the approach are drawn.

Degree: Doctor of Philosophy
Title: *A Type Theory Approach to the Specification and Synthesis of Database Models*
Author: P. Rajagopalan
Institution: The University of Western Australia
Department: Department of Computer Science

In this thesis, we introduce a new approach to the development of database theory that integrates the specification, verification, and implementation of data model structures and functions. As the computational paradigm of database systems is based on the manipulation of data collections, we have developed a generic representation of data collections and a generic algebra for manipulating them. A collection is represented as consisting of elements that satisfy a constraint. This generic construct is used to derive collections that are commonly found in database systems.

The method used in this thesis is based on intuitionistic type theory and constructive logic. Type theory supports the paradigm of proofs as programs where the specification of a program is stated as a theorem, and by proving the theorem the corresponding program can be constructed automatically. An algebra to manipulate generic collections is derived using type theory. Each operator of the algebra is specified as a theorem in constructive logic which describes the nature of the particular operation. A proof of the theorem is derived using a proof development system that implements type theory. From the proof the corresponding

***Abstracts of the Australian and New Zealand Universities
Postgraduate Research Theses***

algebraic operator is extracted as a polymorphic function. Algebras for specific collection constructs are derived from the generic algebra. We show that the generic collection type and its algebra can be used to represent the structures of record-oriented, complex object, and semantic data models, and to manipulate data in those models.

The idempotence, commutativity, associativity, and distributivity properties of the generic algebra are described, and transformation rules based on these properties are formally verified. The generic algebra can be considered as a generalisation of relational algebra. Similarly the generic transformation rules generalise the corresponding rules of relational algebra. The generic rules can be used to derive rewrite rules for algebras of specific data models.

The research reported in this thesis may be viewed as a new approach to the prototyping of database concepts. The advantages and limitations of this approach are discussed. We also discuss the application of our research to heterogeneous database systems (HDBS), by proposing an alternative approach to representing the global schema of a tightly coupled HDBS. The benefits of our approach to the development of database concepts and the scope for further work are also described in the thesis.

Degree:	<i>Master of Engineering Science</i>
Title:	<i>Pattern Classification Image Restoration and Image Coding Using Neural Networks</i>
Author:	<i>Sharad Thacore, B.E. (Hons)</i>
Institution:	<i>The University of Western Australia</i>
Department:	<i>Department of Electrical and Electronic Engineering, School of Information Technology and Electrical Engineering</i>
Awarded:	<i>July, 1993</i>

The highly useful properties of learning, recalling and categorising information has enabled Neural Networks (NNs) to enjoy an extensive application to a wide variety of problems. The particular problems investigated in this thesis, however, are unsupervised pattern classification, image restoration, image block coding and image transform coding. These problems demonstrate the NNs' ability to offer time-efficient solutions. Such efficiency is achieved via the parallel processing and non-linear dynamics of the particular NNs chosen.

Within this thesis, unsupervised pattern classification, using Adaptive Resonance Theory (ART), is studied to illustrate the ART networks' ability to retain coded memories when presented with novel learning material. The need to develop ART software for use by the NN Group within the Department of Electrical and Electronic Engineering at the University of Melbourne, Australia, inspired and investigation of the ART networks. Through this investigation a novel NN, named the Self Organising network with Adaptive Thresholds (SOAT), was developed. The SOAT network has been consequently applied to image block coding.

The problem of image block coding requires good codebook design, which can be achieved by using a well accepted technique, known as the Linde-Buzo-Gray (LBG) algorithm. Research has shown Kohonen's Self Organising Map (SOP) is able to achieve comparable codebook design whilst requiring less computational effort. The SOM requires several presentations of the image data in order to achieve its codebook design. This can become computationally expensive if the number of image sub-blocks is large and the desired codebook is also large. A novel NN, the SOAT network, which is capable of producing equal quality codebooks as the SOM whilst only requiring a single presentation of the image date, is proposed in this thesis as a means of further solving this problem.

Optimisation problems can also be successfully approached using NNs. The discrete and continuous Hopfield networks are applied for this purpose. The application of the discrete Hopfield network for image restoration provides and understanding of the dynamics and stability demonstrated by these networks. The continuous Hopfield network is then used to perform a two dimensional Gabor transform.

The two dimensional Gabor transform of an image provides for excellent compression and segmentation, however, as the Gabor set of functions are non-orthogonal the transform is difficult. Existing solutions to this problem are generally computationally time intensive. A continuous Hopfield network is applied, in the thesis in a novel manner, to perform the Gabor transform using fewer iterations. This is achieved by exploiting the Hopfield network's ability to solve optimisation problems.

Abstracts of the Australian and New Zealand Universities Postgraduate Research Theses

Thus the two main achievements of the thesis, the SOAT network and the novel application of the Hopfield network to perform a Gabor transform, illustrate the legitimate use of NNs for image processing. This work contributes to the fast growing acceptance of NNs as efficient processing tools.

Degree:	<i>Master of Engineering Science</i>
Title:	<i>Interleaving Techniques For High Speed Data Transmission</i>
Author:	<i>Wing Hong Hui</i>
Institution:	<i>The University of Melbourne</i>
Department:	<i>Department of Electrical and Electronic Engineering, School of Information Technology and Electrical Engineering</i>
Awarded:	<i>October, 1993</i>

Interleaving is a technique used to convert a transmission channel with memory into one that is memoryless. The performance of Forward Error Correction (FEC) systems operating in the presence of burst errors is improved by passing the coded signal through an interleaving process. Commercial FEC sub-systems such as Viterbi and Reed-Solomon decoders are now commonplace, however interleavers, while indispensable, are still quite rare.

This dissertation provides a comprehensive review of the two main interleaver types: block and convolutional interleavers. Following this review, the optimum convolutional interleaver is chosen for further analysis. To gain some "real-time" experience and to investigate the commercial potential of a convolutional interleaver, a variable rate interleaver has been successfully implemented on a TMS320C51 Digital Signal Processor (DSP). Many factors were considered in this implementation: throughout, synchronisation, interleaving depth and full-duplex interleaving was finally interfaced to a commercial 1024 QAM 2Mbits/s modem.

The investigation of the implementation of interleavers with DSP indicates that there is a need for more compact and flexible interleaver structures which can be readily integrated (in VLSI or DSP). The final part of the dissertation focused on cascaded and adaptive interleavers. Cascaded interleavers allow more sophisticated interleavers to be constructed from simple interleaving blocks. Adaptive interleavers provide the ability to adjust the interleaving depth (and thus the burst error protection) dynamically. A comprehensive computer simulation was developed and used for these investigations. The previously mentioned DSP based interleaver was also interfaced to the host personal computer (PC). This system facilitates rapid simulation results with the interleaving part of the simulation being run in real time.

In summary, this thesis provides new designs and associated implementation results for various interleaving systems including high speed single chip, variable rate, byte oriented convolutional interleavers. Based on a novel dynamic interleaver concept, a new adaptive interleaving system is proposed and this is supported with successful simulation results for advanced high speed data transmission system.

Degree:	<i>Master of Cognitive Science</i>
Title:	<i>Optical Character Recognition Using Neural Network</i>
Author:	<i>David T. Au</i>
Institution:	<i>The University of Melbourne</i>
Department:	<i>Department of Electrical and Electronic Engineering, School of Information Technology and Electrical Engineering</i>
Awarded:	<i>September, 1993</i>

Handwritten character recognition has been one of the most challenging shape recognition application, because of the large variations in writing style from author to author. Traditional handwritten character recognition systems have employed feature extraction techniques which are tailored to describe the structural and topological characteristics of the characters. Although recognition rates from low to mid 90s can be obtained with these approaches, they are computationally intensive and therefore require sophisticated computing equipment's to operate at an acceptable speed.

This thesis proposes a simple OCR scheme that uses the statistical distribution of points to recognize machine printed and handwritten characters irrespective of translational and rotational distortions. The OCR system

***Abstracts of the Australian and New Zealand Universities
Postgraduate Research Theses***

comprises two stages of operation: feature extraction and classification. In the feature extraction stage, the binary image of the character is transformed to a complex-log map, from which the feature vector is constructed using a partitional projection algorithm. In the classification stage, the feature vectors are used to train a multi-layered feed-forward neural network using the back-propagation algorithm, which minimizes the mean square output error by iteratively adapting the connection weights. The OCR system is used to recognize alphanumerical characters in six Macintosh fonts, and also handwritten characters selected from the CEDAR image database. The results show recognition rates comparable to convention character recognition systems.

Degree:	<i>Doctor of Philosophy</i>
Title:	<i>Robust Variable Structure Control For Robotic Manipulators</i>
Author:	<i>Man Zhihong</i>
Institution:	<i>The University of Melbourne</i>
Department:	<i>Department of Electrical and Electronic Engineering, School of Information Technology and Electrical Engineering</i>
Awarded:	<i>February, 1993</i>

This thesis examines a number of robustness issues of variable structure control for nonlinear robotic manipulators. In particular, this thesis presents robustness and convergence results for variable structure model reference adaptive control systems of robotic manipulators based on system matrix bounds rather than on the upper and the lower bounds of all unknown system parameters. It is also shown that the established results can be extended to the robust decentralised variable structure model reference adaptive control in order to simplify the controller design.

A robust variable structure control scheme based on the nominal system model and the matching conditions of system uncertainties is proposed for robotic manipulators. Using this scheme, the effects of system uncertainties are compensated by using a variable structure compensator in the feedback control system. Robustness with respect to large system uncertainties and asymptotic convergence of the output tracking error can then be obtained.

A generalised robust variable structure control scheme without requirement of the uncertainty matching conditions is developed for robotic manipulators. Moreover, the achieved results are extended to the robust decentralised variable structure control for robotic manipulators by the use of the local information.

A robust variable structure saturation control scheme using the structural properties of robotic manipulators is developed. It can be seen that only five controller parameters are adjusted based on uncertain bounds and strong robustness and asymptotic convergence can then be guaranteed. The results are also extended to the robust decentralised variable structure saturation control for robotic manipulators.

Finally, two robust decentralised nonlinear sliding mode control schemes are provided for robotic manipulators in order to get faster, highly insensitive and more accurate transient error response.

Degree:	<i>Doctor of Philosophy</i>
Title:	<i>Exploring and Utilising Features in Natural Images</i>
Author:	<i>Y. K. Aw</i>
Institution:	<i>The University of Western Australia</i>
Department:	<i>Department of Computer Science</i>
Awarded:	<i>August 1994</i>

Edge detection, or more generally, feature detection, has been widely known as an important process in early vision analysis. The phase congruency feature model, proposed by Morrone and Owens in 1987, predicts the accurate location of features in images perceived by the human visual system. To achieve efficient computation, the phase congruency model is implemented using the local energy model [Venkatesh 1990]. This thesis has shown the discrepancy between the two models and has concluded that the local energy model is a Taylor approximation of the phase congruency model. The conditions which the 1-dimensional implementation of both the phase congruency model and the local energy model must satisfy to achieve

Abstracts of the Australian and New Zealand Universities Postgraduate Research Theses

stability in the location of feature points with respect to the chosen pixel grid orientation in an image are also studied in this thesis.

Unlike the phase congruency model and the local energy model which concentrate on locating feature points, this thesis takes the detection of image features a further step by examining the luminance behaviour of a feature in the local neighbourhood of the point detected by the models. A neural network has been devised to perform the analysis. Various results are obtained. A graded step feature has been found to be the predominant feature in most images. Features are also similar across a class of images and across spatial scales. While not all feature types are present at all scales of an image, features of a more complex image are found to adequately represent features of a simpler image.

The last item of the above-mentioned results has a useful application in the area of image compression and reconstruction. Pointers to a common catalogue of features are maintained for every feature point of an image, and reconstruction is done *via* the coded pointers. The encoded features are found to contain such rich information that even the smooth, featureless regions can be reasonably reconstructed through the propagation of the luminance function from the feature regions. Comparisons with the well-known JPEG image compression technique have been made. The results show the different emphases of the two techniques: the compression technique proposed in this thesis concentrates its resources in the feature regions whereas the JPEG method tries to achieve overall minimum error rate. Therefore the compression ratio attainable by the method proposed in this thesis depends on the complexity of features in an image. In an (almost) extreme case where the image contains only a small black character in the centre of a white background of 256x256 pixels, this method achieves a compression ratio of 182:1 with error free reconstruction at the character's features, while the JPEG method obtains a ratio of 56:1 with 33% of error at the features.

Degree:	<i>Doctor of Philosophy</i>
Title:	<i>Surface Modelling and Surface Following for Robot Equipped with Range Sensors</i>
Author:	<i>Chris Pudney</i>
Institution:	<i>The University of Western Australia</i>
Department:	<i>Department of Computer Science</i>
Awarded:	<i>August 1994</i>

This thesis presents techniques for modelling the surface of an object from data collected by sensors mounted on a robot. The robot moves along the object's surface such that its sensors can take surface readings. Surface modelling takes place during this surface following motion such that the model can be used in the control of surface following. Thus the surface model is constructed incrementally from sets of sensor points obtained during surface following.

A planar polyhedral surface model is introduced that is amenable to incremental surface modelling. Techniques are presented for combining sensor points with surface model segments. The surface model includes a neighbour relation between model segments such that neighbouring segments may share edges. These shared edges give rise to corner vertices. Techniques for the formation and maintenance of shared edges and corners are presented. Also algorithms for combining near-parallel model segments are presented. Finally, the treatment of fissures in model segment perimeters is presented.

The robot's sensors must be moved along the surface such that the surface remains within sensing range but not so closely that the robot collides with the surface. By moving a control point on the robot along the surface model at the fixed distance from the model, those sensors around the control point are kept within range of the surface and the control point is kept from colliding with the surface. The remainder of the robot body is kept from collision by using redundant degrees-of-freedom. By choosing a motion that directs the control point towards incomplete parts of the surface model the sensors are exposed to unexplored sections of the surface model thus contributing to the completion of the surface model.

These techniques have been implemented and tested for a variety of simulated robots and object surfaces and for a U.M.I. RTX robotic manipulator arm equipped with range sensors. The results are presented.

Calendar

Seventh Australian Joint Conference on Artificial Intelligence (AI'94)

Conference date: 21 - 25 November 1994

Conference theme: "Sowing the seeds for the future"

Venue: Armidale, NSW, Australia

Further information:

AI'94 Secretary
 Department of Mathematics, Statistics and Computing Science
 The University of New England
 Armidale, NSW 2351
 Australia
 email: ai94@fermat.une.edu.au

Australian Telecommunication Networks and Applications Conference (ATNAC'94)

Conference date: 5 - 7 December 1994

Venue: Melbourne, VIC, Australia

Further information:

Margaret Keegel
 Conference Manager
 Office of Continuing Education
 Monash University
 PO Box 197
 Caulfield East 3145
 tel: +61 3 903 2808
 fax: +61 3 903 2805
 email: mkeegel@monu6.cc.monash.edu.au

Second Australian and New Zealand Conference on Intelligent Information Systems (ANZIIS-94)

Tutorial date: 29 November 1994

Conference date: 30 November - 2 December 1994

Venue: Brisbane, QLD, Australia

Further information:

ANZIIS-94 Secretariat
 School of Computing Science
 Queensland University of Technology
 GPO Box 2434
 Brisbane, Q 4001
 Australia
 tel: +61 7 864 2925
 fax: +61 7 864 1801
 email: anziis94@qut.edu.au

Seiken/IEEE Symposium on Emerging Technologies & Factory Automation (EFTA94)

Conference date: 6 - 10 November 1994

Conference theme: "Novel Disciplines for Next Century"

Venue: Tokyo, Japan

Further information:

Richard Zurawski
 Laboratory for Robotics & Intelligent Systems
 Department of Elec & Comp Engineering
 Swinburne University of Technology
 PO Box 218 Melbourne VIC 3122
 Australia
 tel: +61 3 728 7161
 fax: +61 3 728 7183
 email: rzz@stan.xx.swin.oz.au

Fifth Australian International Conference on Speech Science and Technology (SST-94)

Tutorial date: 5 December 1994

Conference date: 6 - 8 December 1994

Venue: Perth, WA, Australia

Further information:

Dr. Roberto Tognoni, Secretary SST-94
 CIIPS, Dept. of E&E Engineering
 The University of Western Australia
 Nedlands, WA 6009
 tel: +61 9 380 2535/3897
 fax: +61 9 380 1101
 email: sst94@ee.uwa.edu.au

International Conference on Neural Information Processing (ICONIP '94)

Conference date: 17 - 20 October 1994

Venue: Seoul, Korea

Further information:

ICONIP'94 - Seoul Secretariat
 C/o Intercom Convention Service, Inc.
 SL. Kang Nam P.O. Box 641
 Seoul 135-606 Korea
 tel: +82 2 515 1560/546-7065
 fax: +82 2 516 4807
 email: ICONIP@cair.kaist.ac.kr

Calendar

Australian Neural Networks Conference

Conference date: 6-8 February 1995

Venue: Sydney, Australia

Deadlines:

paper submission: 2 September 1994
notification of acceptance: 28 October 1994

Further information:

Agatha Shotam
ACNN'95 Secretariat
University of Sydney
Department of Electrical Engineering
NSW 2006

International Conference on Open Distributed Processing (ICODP)

Conference date: 20-24 February 1995

Venue: Brisbane, Australia

Further information:

ICODP'95
DSTC
Level 7, Gehrmann Laboratories
University of Queensland
QLD 4072, Australia
tel: +61 7 365 4310
fax: +61 7 365 4311
email: icodp95@tdstc.edu.au

ACSC'95 and ADC'95 18th Australasian Computer Science Conference & 6th Australasian Database Conference

ADC'95 Conference date: 30-31 January 1995

ACSC'95 Conference date: 1-3 February 1995

Venue: Adelaide, Australia

Deadlines:

paper submission: 19 August 1994
notification of acceptance: 4 November 1994

Further information:

ACSC'95 or ADC'95
Discipline of Computer Science
Flinders University
GPO Box 2100, Adelaide
South Australia, 5001

IEEE International Conference on Fuzzy Systems FUZZ-IEEE/IFES'95

Conference date: 20 - 24 March 1995

Venue: Yokohama, Japan

Further information:

Prof. Toshio Fukuda
Nagoya University
Dept. of Mechanical Engineering
Furo-cho, Chikusa-ku, Nagoya 464-01, Japan
tel: +81 52 781 5111 x4478
fax: +81 52 781 9243
email: d43131a@nucc.cc.nagoya-u.ac.jp

The Australian Society for Operation Research 13th Conference

Conference date: 25-27 September 1995

email: dlh@cs.adfa.oz.au

5th International Conference on Telecommunications Information Networking Architecture TINA'95

Conference date: 13-16 February 1995

Venue: Melbourne, Australia

Deadlines:

paper submission: 23 September 1994
demo proposals: 23 September 1994
video proposals: 23 September 1994
notification of acceptance: 1 November 1994

Further information:

TINA'95
Convention Network
224 Rouse Street
Port Melbourne, VIC 3207
Australia
tel: +61 3 646 4122
fax: +61 3 646 7737
email: tina95@trlo.oz.au

The Australasian Conference for Group Systems Technology

Conference date: 27-29 September 1995

email: leone@cs.adfa.oz.au

Calendar

IEEE First International Conference on Algorithms and Architectures for Parallel Processing ICA3PP-95

Conference date: 19-21 April 1995

Venue: Brisbane, Australia

Deadlines:

paper submission:	15 October 1994
tutorial proposals:	1 November 1994
notification of acceptance:	4 November 1994

Further information:

Dr. V.L.Narasimhan
 The PA3SE Research Laboratory
 Dept of Elect. & Comp. Engineering
 The University of Queensland
 Qld 4072, Australia
 tel: +61 7 365 3573
 fax: +61 7 365 4999
 email: ica3pp95@elec.uq.oz.au

The 2nd New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems ANNES'95

Conference date: 20-23 November 1995

Venue: University of Otago, Dunedin, New Zealand

Deadlines:

paper submission:	30 May 1995
notification of acceptance:	10 July 1995
final copy:	20 August 1995

Further information:

ANNES'95 Secretariat
 Dept of Information Science
 University of Otago
 P.O. Box 56
 Dunedin
 New Zealand
 email: annes95@otago.ac.nz

8th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems IEA'95 AIE

Conference date: 5-9 June 1995

Venue: Melbourne, Australia

Deadlines:

paper submission:	18 November 1994
notification of acceptance:	22 January 1995
final copy:	22 February 1995

Further information:

IEA'95 AIE
 Dr Moonis Ali
 Dept of Computer Science
 Southwest Texas State University
 San Marcos TX 78666-4616
 USA
 email: ma04@academia.swt.edu

1995 IEEE Workshop on Nonlinear Signal and Image Processing

Conference date: 20-22 June 1995

Venue: Halkidiki, Greece

Deadlines:

paper submission:	1 November 1994
notification of acceptance:	10 January 1995
final copy:	1 March 1995

Further information:

Prof Y. Attikiouzel
 CIIPS
 The University of Western Australia
 Nedlands, WA 6009
 Australia
 email: yianni@ee.uwa.edu.au

Calendar

Second Australian and New Zealand Conference on Intelligent Information Systems (ANZIIS-94)

30 November - 2 December 1994

Tutorials: 29 November 1994

Brisbane, Queensland

There are four targeted areas:

- artificial intelligence*
- fuzzy systems*
- neural networks*
- evolutionary computation*

Papers are requested from, but not limited to, the following areas:

- adaptive systems
- artificial life
- autonomous vehicles
- data analysis
- factory automation
- financial markets
- intelligent databases
- knowledge engineering
- machine vision
- pattern recognition
- neurobiological systems
- control systems
- optimisation
- parallel & distributed computing
- robotics
- prediction
- sensorimotor systems
- signal processing
- speech processing
- virtual reality

For further information see the calendar entry in this issue.

Fifth Australian International Conference on SPEECH SCIENCE and TECHNOLOGY SST-94

6 - 8 December 1994

Perth, Western Australia

The SST-94 conference is organized with a multidisciplinary perspective to provide excellent opportunities for the exchange of ideas and facilitate the interaction between professionals from many diverse areas of expertise. General topics of interest are, but not limited to:

- Speech Synthesis
- Speech Recognition
- Text-to-Speech Synthesis
- Acoustic Phonetics and Prosody
- Speech Signal Analysis
- Speech Databases
- Voice Response Systems
- Human Audition, Perception and Cognition
- Speech Processing using AI, ANN and advanced techniques
- Speech Coding and Encryption
- Spoken Language Modelling
- Speech Production
- Speaker Characteristics
- Speaker Identification/Verification
- Speech Disorders
- Human-Machine Speech Interfaces
- Aids for the Speech/Hearing Impaired
- Speech Technology Applications

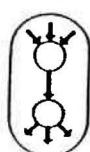
This ensures that SST-94 is a true reflection of the interdisciplinary nature of speech as an area of scientific and industrial endeavour. A tutorial day will be held on Monday, December 5 as an introduction to the important concepts in speech science and technology research. Keynote addresses will be given by Prof. Bob Lingard from the University of East Anglia and Dr. Anne Cutler from the Max-Planck-Institute in the Netherlands.

For further information see the calendar entry in this issue.



IEEE

1995 IEEE INTERNATIONAL CONFERENCE ON EVOLUTIONARY COMPUTING (ICEC'95)



29 November - 1 December 1995
Perth, Western Australia



General Chair
Yianni Attikiouzel
University of Western Australia
Perth, Australia

Technical Chair
David B. Fogel
Natural Selection, Inc.
La Jolla, USA

Assistant Technical Chair
Ah Chung Tsoi
University of Queensland
St. Lucia, Australia

International Liaison Chair
Toshio Fukuda
Nagoya University
Nagoya, Japan

European Liaison Co Chairs
Marco Dorigo
Universite' Libre de Bruxelles
Brussels, Belgium

Reinhard Maenner
Universitaet Mannheim
Mannheim, Germany

Treasurer
Svetna Venkatesh
Curtin University of Technology
Perth, Australia

Local Arrangements
Dorota Kieronska
Curtin University of Technology
Perth, Australia

Publicity Chair
Chris deSilva
University of Western Australia
Perth, Australia

CALL FOR PAPERS

The IEEE International Conference on Evolutionary Computing will be concerned with the theory and applications of genetic and evolutionary algorithms, and associated topics. It will be held concurrently with the last three days of ICNN'95. Submissions of papers related to the topics listed below are invited.

TOPICS

Theory of evolutionary computation
Applications of evolutionary computation
Efficiency/robustness comparisons with other direct search algorithms
Parallel computer implementations
Artificial life
Evolutionary algorithms for computational intelligence
Comparisons between different variants of evolutionary algorithms
Machine learning applications
Evolutionary computation for neural networks
Fuzzy logic in evolutionary algorithms

SUBMISSION PROCEDURE

Prospective authors are invited to submit papers related to the listed topics for oral or poster presentation. Five (5) copies of the paper must be submitted for review. Papers should be printed on ISO A4 or USA 8.5" x 11" white paper, written in English in one-column format in Times or similar font style, 10 points or larger with 2.5cm (1") margins on all four sides. A length of four pages is encouraged, and a limit of six pages, including figures, tables and references will be enforced.

Centered at the top of the first page should be the complete title and the name(s), affiliation(s) and address(es) of the author(s). The *Paper Submission Cover Sheet* on the back of this *Call for Papers* should be completed and attached to the submission. Papers should be sent to:

David B. Fogel
Natural Selection, Inc.
1591 Calle De Cinco
La Jolla, CA 92037
USA

ADDRESS

All correspondence regarding the conference, other than submissions of papers, should be addressed to:

ICEC'95 Conference Management
Centre for Intelligent Information Processing Systems
The University of Western Australia
Nedlands WA 6009
AUSTRALIA
Telephone: +61 9 380 1969 Fax: +61 9 380 1101
Electronic mail: ec95@ee.uwa.edu.au

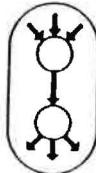
SCHEDULE

Proposals for tutorials, exhibits and plenaries:	31 March 1995
Submission of papers:	5 May 1995
Notification of acceptance:	7 July 1995
Submission of camera-ready papers:	25 August 1995



1995 IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS (ICNN'95)

27 November - 1 December 1995
Perth, Western Australia



General Chair
Yianni Attikiouzel
*University of Western Australia
Perth, Australia*

Technical Program Co-Chairs
Marimuthu Palaniswami
*University of Melbourne
Melbourne, Australia*

Toshio Fukuda
*Nagoya University
Nagoya, Japan*

Robert J Marks III
*University of Washington
Seattle, Washington, USA*

International Liaison Chair
Shun-ichi Amari
*University of Tokyo
Tokyo, Japan*

US Liaison
Jim Bezdek
*University of West Florida
Pensacola, USA*

European Liaison
Peter Noakes
*University of Essex
Colchester, UK*

Treasurer
Svetha Venkatesh
*Curtin University of Technology
Perth, Australia*

Local Arrangements
Dorota Kieronska
*Curtin University of Technology,
Perth, Australia*

Publicity Chair
Chris deSilva
*University of Western Australia
Perth, Australia*

CALL FOR PAPERS

The IEEE International Conference on Neural Networks will be concerned with natural and artificial neural networks. Submissions of papers related to the topics listed below are invited.

TOPICS

Applications	Motion Analysis
Architectures	Neurobiology
Artificially Intelligent Neural Networks	Neurocognition
Associative Memory	Neurodynamics
Computational Intelligence	Optimization
Cognitive Science	Pattern Recognition
Filtering	Prediction
Fuzzy Neural Systems	Robotics
Hybrid Systems	Sensation and Perception
Image Processing	Sensorimotor Systems
Implementations	Speech, Hearing and Language
Intelligent Control	System Identification
Learning and Memory	Supervised/Unsupervised Learning
Machine Vision	Time Series Analysis

SUBMISSION PROCEDURE

Prospective authors are invited to submit papers related to the listed topics for oral or poster presentation. Five (5) copies of the paper must be submitted for review. Papers should be printed on ISO A4 or USA 8.5" x 11" white paper, written in English in one-column format in Times or similar font style, 10 points or larger with 2.5cm (1") margins on all four sides. A length of four pages is encouraged, and a limit of six pages, including figures, tables and references will be enforced.

Centered at the top of the first page should be the complete title and the name(s), affiliation(s) and address(es) of the author(s). The *Paper Submission Cover Sheet* on the back of this *Call for Papers* should be completed and attached to the submission. Papers should be sent to the address below.

ADDRESS

All correspondence regarding the conference should be addressed to:

ICNN'95 Conference Management
Centre for Intelligent Information Processing Systems
The University of Western Australia
Nedlands WA 6009
AUSTRALIA
Telephone: +61 9 380 1969 Fax: +61 9 380 1101
Electronic mail: icnn95@ee.uwa.edu.au

SCHEDULE

Proposals for tutorials, exhibits and plenaries:	31 March 1995
Submission of papers:	5 May 1995
Notification of acceptance:	7 July 1995
Submission of camera-ready papers:	25 August 1995

Australian Journal of Intelligent Information Processing Systems

INFORMATION FOR AUTHORS

Camera-ready copy

Australian Journal of Intelligent Information Processing Systems (AJIIPS) is published quarterly. The aim of AJIIPS is to publish papers describing theory, methods and techniques, applications and tutorial presentations in a range of areas relating to computer engineering and computer science. AJIIPS publishes full papers, short notes and survey articles. Specific areas include but are not limited to: Artificial Intelligence, Artificial Neural Networks, Computer Science, Fuzzy System and Virtual Reality.

All papers will be reviewed by at least two reviewers and one of the editors. Decision as to the suitability of the paper will be made by the Editor in Charge.

Papers should be clearly presented, consistent with giving proper description of the primary contribution. Theory papers should be based on clear, formal foundations; methods and techniques papers should indicate the novelty and advantage of the technique; application papers should present appropriate results and evaluation of performance; tutorial papers should be clearly presented to a reader with a general background in all areas of interest but no prior background in the specific topic.

Submission of technical papers

Manuscripts must be written in English. The papers should be original work, not appearing in any other journal, although extended versions of conference papers may be considered. The authors are expected to obtain all relevant copyright releases for any copyrighted material included in their paper.

Potential authors should send to the Editor in Chief 5 (five) copies of the complete manuscript. The paper, including an abstract, should not exceed 10 pages. To allow for anonymous refereeing, a separate page should be included containing the authors' names, affiliations, including email address, manuscript's title and the abstract. Authors' names and affiliations should not appear on the manuscript. Submissions by more than one author should indicate the author to whom the correspondence will be addressed.

At the time of acceptance, the authors will be requested to provide a technical biography and a photograph of each author of the paper (except for Short Notes). The final, camera-ready document should conform to the presentation standard set out below. The first page should include the title in Helvetica, 18 point, bold, and authors' names and affiliation in Times, 12 point, bold and italics (see sample page).

There will be no page charges for published papers.

Manuscript preparation

Submissions, in camera-ready form, should conform to the following rules:

- 1) laser-printer quality on one side of A4 paper with margins: left and right - 1.8 cm, top - 2.3 cm, bottom - 1.8 cm.
- 2) text should be in Times Roman, 10 point font, in two column layout with 0.63 cm gap
- 3) headings should be left-justified and use Helvetica font varying as follows:
 - 1 Level One Heading - 14 point bold
 - 1.1 Level Two Heading - 12 point bold
 - 1.1.1 Level Three Heading - 10 point bold
 - 1.1.1.1 Level Four Heading - 10 point italics

- 4) A short abstract should be provided, 100 - 200 words, in one column, 10 point Times Roman font, in italics, with the abstract keywords in bold. Note that short papers (or short notes) require a shorter abstract of up to 50 words.
- 5) Footnotes should be numbered and should appear, in 9 point Times Roman font, at the bottom of respective columns.
- 6) Standard abbreviations should be used if possible, and nonstandard abbreviations must be defined before being used. All units of measurement should be metric.
- 7) Originals for illustrations should be clear and of good quality. Figures should use centred captions in 10 point Times Roman font, with the words Figure 1 in bold (see sample page). Tables should be numbered using upper case Roman numerals, with the table heading (in the same format as for figures) appearing below the table.
- 8) References should appear as the last section at the end of the paper. They should be sorted by author, and numbered with Arabic numerals in square brackets [1] - see the sample page.

Style for papers: author(s) - surname separated by a comma, followed by the initials, title (between double quotes), journal title, volume, inclusive page numbers, month and year.

Style for books: author(s), title, location, publisher, year, chapter or page numbers (as appropriate).

SUBSCRIPTION ORDER FORM

Australian Journal of Intelligent Information Processing Systems
AJIIPS - ISSN 1321-2133

Fill in, detach and return to:

Prof. Yianni Attikiouzel
CIIPS
The University of Western Australia
Nedlands, WA 6009 Australia

- Individual rate:
 \$60 within Australia \$70 outside Australia
- Institutional rate:
 \$200 within Australia \$230 outside Australia
- Please send me a sample copy
- I enclose a cheque or money order for \$_____

I enclose a cheque or money order for \$_____

Charge my: Master Card VISA
Credit Card # _____

Name on the card: _____

Expiry date: _____

Signature: _____

Send journal to:

Name: _____

Address: _____

Please note: Subscriptions are entered upon receipt of payment. Four issues starting with the next publication will be shipped.

Cover design by Jill Smith, Artist-in-residence, Department of Computer Science, Curtin University of Technology, Perth, Western Australia.

Australian Journal of Intelligent Information Processing Systems

ISSN: 1321-2133

Volume 1, No. 2

June 1994

CONTENTS

ARTIFICIAL NEURAL NETWORKS

On Two Methods of Accelerating Learning in Feedforward Networks <i>R.J. Gaynier, T. Downs</i>	1
Efficient Multichannel Image Recognition Using Feedforward Multiresolution Neural Networks <i>D. Kalogerias, S. Kollias</i>	8
A Back-Propagation Adaptation Yielding a Ternary Output ANN with Limited Integer Weights <i>G.R. Cukowicz, M.R. Haskard</i>	16

ARTIFICIAL INTELLIGENCE

A Comprehensive Synthesis Strategy for Conflict Resolutions in Distributed Expert Systems <i>M. Zhang, C. Zhang</i>	21
Family Law, Natural Language Understanding and AI <i>W.K. Yeap</i>	29

GENETIC ALGORITHMS

An Introduction to Evolutionary Computation <i>D.B. Fogel</i>	34
--	----

IMAGE PROCESSING

A Modified Lagrange's Interpolation for Image Reconstruction <i>S. Suthaharan</i>	43
--	----

SELECTED AUTHOR BIOGRAPHIES

53

ABSTRACTS OF THE AUSTRALIAN UNIVERSITIES POSTGRADUATE RESEARCH THESES

54

CALENDAR

66

