

AD-A234 886



**RADC-TR-90-404, Vol VII (of 18)
Final Technical Report
December 1990**



2

AUTOMATIC PHOTOINTERPRETATION

Northeast Artificial Intelligence Consortium (NAIC)

James Modestino and Arthur Sanderson

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

This effort was funded partially by the Laboratory Director's fund.

**Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700**

This report has been reviewed by the RADC Public Affairs Division (PA) and is releasable to the National Technical Information Services (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

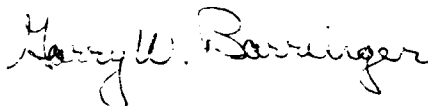
RADC-TR-90-404, Volume VII (of 18) has been reviewed and is approved for publication.

APPROVED:



LEE A. UVANNI
Project Engineer

APPROVED:



GARRY W. BARRINGER
Director of Special Programs
Directorate of Intelligence & Reconnaissance

FOR THE COMMANDER:



IGOR G. PLONISCH
Directorate of Plans & Programs

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRRE) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE December 1990	3. REPORT TYPE AND DATES COVERED Final Sep 84 - Dec 89	
4. TITLE AND SUBTITLE AUTOMATIC PHOTOINTERPRETATION			5. FUNDING NUMBERS C - F30602-85-C-0008 PE - 62702F PR - 5581 TA - 27 WU - 13 (See reverse)	
6. AUTHOR(S) James Modestino and Arthur Sanderson			8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Northeast Artificial Intelligence Consortium (NAIC) Science & Technology Center, Rm 2-296 111 College Place, Syracuse University Syracuse NY 13244-4100			10. SPONSORING/MONITORING AGENCY REPORT NUMBER RADC-TR-90-404, Vol VII (of 18)	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Rome Air Development Center (CCGS) Griffiss AFB NY 13441-5700			(See reverse)	
11. SUPPLEMENTARY NOTES RADC Project Engineer: Lee A. Uvanni/IRRE/(315) 330-4863 This effort was funded partially by the Laboratory Director's fund.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The Northeast Artificial Intelligence Consortium (NAIC) was created by the Air Force Systems Command, Rome Air Development Center, and the Office of Scientific Research. Its purpose was to conduct pertinent research in artificial intelligence and to perform activities ancillary to this research. This report describes progress during the existence of the NAIC on the technical research tasks undertaken at the member universities. The topics covered in general are: versatile expert system for equipment maintenance, distributed AI for communications system control, automatic photointerpretation, time-oriented problem solving, speech understanding systems, knowledge base maintenance, hardware architectures for very large systems, knowledge-based reasoning and planning, and a knowledge acquisition, assistance, and explanation system. The specific topics for this volume are the use of expert systems for automated photo interpretation and other AI techniques to image segmentation and region identification.				
14. SUBJECT TERMS Artificial Intelligence, Expert Systems, Machine Vision, Photointerpretation, Image Analysis			15. NUMBER OF PAGES 48	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

Block 5 (Cont'd)

Funding Numbers

PE - 62702F	PE - 61102F	PE - 61102F	PE - 33126F	PE - 61101F
PR - 5581	PR - 2304	PR - 2304	PR - 2155	PR - LDFP
TA - 27	TA - J5	TA - J5	TA - 02	TA - 27
WU - 23	WU - 01	WU - 15	WU - 10	WU - 01

Block 11 (Cont'd)

This effort was performed as a subcontract by Rensselaer Polytechnic Institute to Syracuse University, Office of Sponsored Programs.

RENSSELAER POLYTECHNIC INSTITUTE
NAIC ANNUAL REPORT
1989

Project: REPRESENTATION AND LEARNING FOR COMPUTER VISION

PARTICIPANTS:

Arthur C. Sanderson, Professor, ECSE Dept.
Michael Skolnick, Professor, Computer Science Dept.
Balakrishnan Ravichandran, Research Assistant
David Jacobs, Research Assistant

PUBLICATIONS:

1. Sanderson, A. C., "Attributed Image Matching Using a Minimum Representation Criterion," Proc. of IEEE 1989 International Conference on Robotics and Automation.
2. Sanderson, A. C., "A Minimum Representation Criterion for Image matching," in review, IEEE Trans. on Pattern Analysis and Machine Intelligence, 1989.

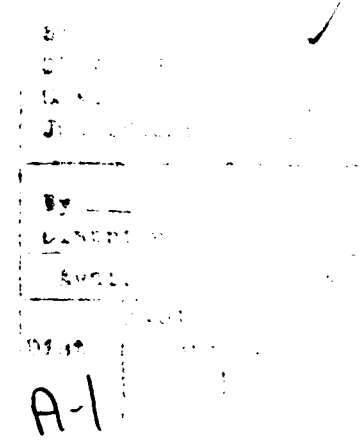
CONFERENCES:

1. Sanderson, A. C., "Attributed Image Matching Using a Minimum Representation Criterion," 1989 International Conference on Robotics and Automation, May 1989.
2. Skolnick, M. M., 3rd International Conference on Genetic Algorithms, June 1989.

TECHNICAL RESULTS:

ATTRIBUTED IMAGE MATCHING USING A MINIMUM.....1
REPRESENTATION SIZE CRITERION

AN APPROACH TO REPRESENTING SPATIAL INFORMATION.....26
INFORMATION USING GENETIC ALGORITHMS
AND CLASSIFIER SYSTEMS



Northeast Artificial Intelligence Consortium

1989 Annual Report

**ATTRIBUTED IMAGE MATCHING USING A
MINIMUM REPRESENTATION SIZE CRITERION**

Arthur C. Sanderson and Balakrishnan Ravichandran

Electrical, Computer, and Systems Engineering Department,
Rensselaer Polytechnic Institute, Troy, NY 12180

1 INTRODUCTION

Matching of models to image features is a fundamental step in computer vision systems. Such matching may take place at different levels of these systems, from template matching of raw images to symbolic matching of relational models. In this report, we address the problem of matching localized spatial features with arbitrary attribute sets to either idealized or learned models. In mathematical terms, we match spatial patterns of points, where each point has an associated attribute vector with quantitative and symbolic values. The minimum representation criterion used to achieve an acceptable match is a principal topic of this report.

Image matching is difficult to achieve with sufficient generality, speed, and robustness to be useful in practical systems. Many proposed algorithms are highly dependent on a choice of particular features and model representation, and they often require interactive or heuristic methods to extract features. Adding generality to matching procedures has been difficult particularly because evaluation functions or match quality measures do not generalize well. Image matching is inherently complex from a computational point of view, since the number of possible matches in general grows exponentially with the number of features. Polynomial complexity is an important property of any practical approach.

Good image matching algorithms must be able to handle feature uncertainty including missing data, extra features, and noisy attributes. This requirement has been particularly difficult to achieve since most evaluation functions are not

able to handle missing or extra data in a consistent non-heuristic fashion. The representation criterion presented in this report is inherently normalized to match size and number of attributes and directly accomodates missing and extra data.

This report describes the minimum representation criterion [1,2,3,4] as a basis for image transformation and correspondence matching. We specifically address the problem of two-dimensional rigid, attributed point sets with missing and extra points. The algorithms developed are polynomial in complexity and near-optimal for this criterion. Examples of performance on highly variable gray-level images including aerial imagery are shown. Results which have been obtained on the application of minimum representation matching techniques to several types of imagery including aerial photographs obtained from RADC are summarized. While the underlying methodology for the minimum representation approach has been developed in [3,4], the current work has emphasized a new implementation of the work and application to new types of imagery. This report includes an overview of the basic methodology, new implementation, and new applications, and augments the papers which have been prepared summarizing our results.

Section 2 of this report defines the image matching problem. Section 3 presents the minimum representation criterion principles. Section 4 describes a usually optimal, polynomial time algorithm for image matching and transformation. Section 5 presents some examples of the matching procedure.

2 ATTRIBUTED IMAGE MATCHING

Image matching problems have been approached using a variety of different hypothesize-and-test techniques in which potential matches are hypothesized and tested against evaluation criteria. These methods include template correlation [5], statistical pattern recognition [5], parameterized geometric fitting [6], and many different relational structure methods such as graph morphisms [7], compatibility graphs [8,9], and weighted relational matching [10]. In addition, heuristic techniques [11], Hough transform techniques [12], and relaxation labelling techniques [13] have been proposed. These references indicate examples of the various approaches, and a more detailed comparative discussion of these algorithms is included in [4]. The approach described in this report is basically a geometric fitting technique which maps point sets to geometric models using a new metric for

evaluating match quality. The minimum representation metric does not depend on the specific form of geometric modelling and is extendible to more general relational structure models.

In this report, we consider images of rigid objects which have undergone arbitrary translation, rotation, and scaling in a two-dimensional plane parallel to the image plane. Each input image of an object is represented as a set of features with attributes, and each object model is represented in a similar manner for a given view of the object. In practice, the input image feature representation is extracted from the raw image data using other computer vision algorithms. The corresponding object model representation may be derived from a purely geometric model or by learning from a series of observations of input images. In addition to translation, rotation, and scaling, the image feature representation will include distortion, noisy attributes, missing (hidden or occluded) features, and added features. The image matching problem requires identification of the correspondence match between features and an associated geometric transformation which 'aligns' the image with the object model. The existence of an arbitrary transformation and the contribution of distortion and noise require a search over possible choices using an evaluation criterion which is tolerant to these effects. In this report, the minimum representation criterion is used for the selection of the best correspondence and transformation.

An input image data feature representation consists of the ordered pair $D = (F, A)$ where $F = \{f_i, i = 1, \dots, N\}$ is the set of feature labels, and $A = \{a_{ij}, i = 1, \dots, N, j = 1, \dots, N_a\}$ is the set of feature attributes.

Each feature may have multiple attributes, and the set of attributes may differ among features. However, every feature in an image is required to have (x,y) position attributes denoted by

$$a_{i1} = u_i = x\text{-position of } f_i,$$

$$a_{i2} = v_i = y\text{-position of } f_i.$$

Similarly, an object model feature representation consists of an order pair $R = (G, B)$ where $G = \{g_i, i = 1, \dots, M\}$ is the set of model feature labels, and $B = \{b_{ij}, i = 1, \dots, M, j = 1, \dots, M_b\}$ is the set of model feature attributes. In this case

$b_{i1} = x_i = x\text{-position of } g_i,$

$b_{i2} = y_i = y\text{-position of } g_i.$

The attributes represented by A and B may be of four types:

1. *Positional* - (x,y) -position (required of every feature),
2. *Numerical* - numerical measures such as length, angle, area, curvature, number of neighbors,
3. *Symbolic* - symbolic labels such as color, texture,
4. *Relational* - relation of a feature to other features such as connected-to, on-top-of.

This data structure considers attributes independently and facilitates the development of the representation criterion which is strictly cumulative with respect to the set of features. For the problems considered in this report, relational attributes will not be used. For highly noisy data relational attributes are difficult to incorporate into matching, and for rigid objects they are less useful since relative position is maintained by the rigid transformation.

2.1 Correspondence

Given an object model R and an input image I with data feature representation D , a match between them is defined by a correspondence and a transformation. The correspondence maps the model features G to the data features F . The transformation is the set of parameters which defines the translation, rotation, and scaling used to geometrically align the corresponding features. In this report, we assume that all correspondence matches are one-to-one, that is, one model feature matches to only one image feature and vice-versa. This assumption may be generalized, but simplifies the search problem and provides solutions which are more easily interpreted.

The *size* of the correspondence match, $N_m \leq \min[M, N]$, is the number of model features which have a correspondence to a designated data feature. Not all model features have matches, and there may be added features in the image as well. The correspondence itself is expressed by the set of indices: $C = \{c_i, i = 1, \dots, M\}$. where

c_i = index of the image feature, f_{c_i} , which corresponds to the indicated model feature, g_i , when a match occurs and,

= 0, when no match occurs,

and $1 \leq c_i \leq N$. A particular correspondence match may therefore be represented by the ordered pair (i, c_i) .

2.2 Transformation

Given a correspondence (i, i) where the model feature g_i is at point (x_i, y_i) and the image data feature f_i is at point (u_i, v_i) , then the match is completely defined by a transformation T which transforms $(u_i, v_i) \rightarrow (x_i', y_i')$. In general, this transformation is defined by four parameters, $T = (t_u, t_v, O, s)$, where

(t_u, t_v) = translation,

O = rotation angle,

s = scaling magnitude.

Fig. (1) illustrates such a transformation from (u_i, v_i) to (x_i', y_i') . While the data point is matched to the model point, the transformed data point does not necessarily align perfectly with the model point. The transformation will be derived from an evaluation criterion over a set of distorted and noisy data points, and will align relative to that global measure.

3 MINIMUM REPRESENTATION CRITERION

The minimum representation criterion [1,2,3] was introduced as an approach to unsupervised signal and data analysis in which the complexity of the data representation is used as a criterion for the choice of model structures and model parameters. The approach incorporates elements which express the complexity of the modeling procedure, the model size, and the size of the data residuals. In contrast to traditional mean square error measures of model fit which do not permit discrimination among model structures, the minimum representation size explicitly incorporates model structure and represents the tradeoff between complexity of the model structure and the resulting error in predicting the data points. This approach was demonstrated for several classes of parametric statistical models including evaluation of the order of an autoregressive model and determination

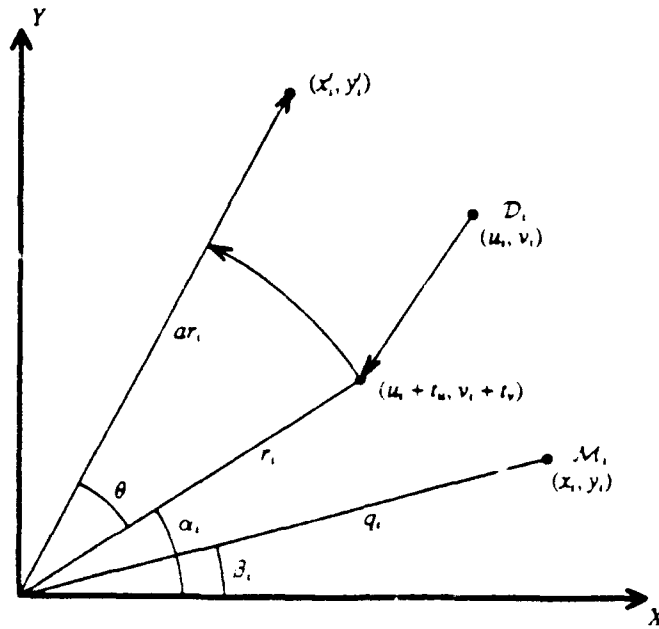


Figure 1 Transformation of an image point at (u, v) to a new point (x', y') using transformation T with four parameters: translation (t_u, t_v) , rotation θ , and scaling s .

of the number of clusters in a multivariate data sample. In [2], these techniques were applied to the unsupervised analysis of biomedical signals which resulted in a system for the automatic modeling, segmentation, and symbolic representation of complex patterns associated with medical diagnostic decisions.

The minimum representation criterion is based on a principle of minimum complexity of a program which explicitly regenerates observed data. Such a program includes a procedure, a model, and data residuals, and the size of the overall program is regarded as a measure of the complexity of the representation. In this approach, a simple model may require a complex data residual representation, while a more complex model will simplify the data residual representation. This tradeoff in overall complexity between model size and data residual size inherently provides a basis for choosing among alternative models. More generally, the procedure provides a tool for unsupervised decision-making.

Consider an observed data vector $\mathbf{x} = [x_1, x_2, \dots, x_N]$. The representation of this data vector is viewed as a program which regenerates the data points with some known resolution. In [1], this program is more formally defined in terms of a classic Turing machine model of computation. There may, in fact, be several different programs, π , which correctly generate the data points, and the 'correct' behavior of the system is regarded as the minimum size program p^* among these,

such that

$$s(p^*) = \min s(p_i) \quad (\text{bits})$$

where $s(\cdot)$ is the size of the program. As discussed in [1], the shortest program in an ensemble of such programs generated by a random process is the most likely program.

Each program, p , includes a number of segments which provide procedure code, model parameters, correspondence parameters, and data residuals. Each different algorithm or different model has a different set of program segments. In our previous work on clustering [1], for example, the model parameters included the cluster center positions in multivariate space, while the data residuals were encoded relative to these centers using a code which minimized the length of the data representation by encoding more probable (closer to the cluster center) data points with shorter length codes. In the image matching problem, the representation size $s(p)$ of each program includes the following terms:

$$s(p) = L + s(q) + s[C_q(\mathbf{x})] + s(\epsilon),$$

where

L = size of the program independent of the choice of model,

$s(q)$ = size of model parameters, including the transformation, the number of modeled data points, N_m , the correspondence match, and the feature attributes,

$s[C_q(\mathbf{x})] = [-\log P_q(\mathbf{x})]$, where

$C_q(\mathbf{x})$ = encoded residuals of modeled points, where

$P_q(\mathbf{x})$ = probability density function of the residuals of the modeled subset of observed data point attributes relative to the model q , and

$$s(e) = \sum_i \sum_j s(a_{ij})$$

is the representation size for the unmodeled data points. When all data points have uniform attribute sets, we can further simplify this to

$$\begin{aligned} s(e) &= (N - N_m) \sum_j s(a_j), \\ &= (N - N_m) S_a. \end{aligned}$$

where S_a is the total representation size for the attributes of each unmodeled data point. In practice, S_a depends on the predefined resolution in bits of each of the attributes and is therefore usually fixed for a given problem.

The representation of the data residuals is based on an encoding which represents the more likely points by shorter code strings. There are many specific coding schemes which might be used, and we have implemented one such scheme which is based on a truncated hyperbolic distribution of errors. Incorporating this measure, we can write the representation size equation for a fixed model and data size in the following form:

$$s(p) = L + N_m \log_2 M + \sum_i s_i + (N - N_m) S_a.$$

where

$$s_i = \sum \log [w_{ij} E_{ij} + 1].$$

E_{ij} is the error due to the j th attribute at feature i ,

$$E_{ij} = \text{Error}_j [g_i, f_{c,i}],$$

and w_{ij} is a weighting parameter which can be used to adjust the relative weight of attributes for different specific applications. For the image matching problem we have used Euclidean error measures as a basis for the encoding of position attributes, and the resulting representation size equation is

$$\begin{aligned} s(p) &= L + N_m \log_2 M + \sum \log_2 \left[\left((x'_i - x_i)^2 + (y'_i - y_i)^2 \right)^{1/2} + 1 \right] + \\ &\quad \sum_i \sum_{j=3}^{N_a} \log_2 [w_{ij} E_{ij} + 1] + (N - N_m) S_a. \end{aligned}$$

For the experiments described in this report the second term $N_m \log_2 M$ was considered a constant for each set of experiments.

4 IMAGE MATCHING ALGORITHM

For a given model and observed data, the best match is defined by an optimal transformation and an optimal correspondence between some subset of the data points and a subset of the model points. These two steps may be considered somewhat independently. An optimal transformation will exist for each possible correspondence which is chosen, and the algorithm must search over many possible correspondences in order to find an optimal match.

4.1 Transformation

The minimum representation transformation is in general quite different from the least mean square error transformation which is commonly used. A closed form analytical expression for the least mean squared error transformation may be derived and applied directly to a given model and subset of data points. The minimum representation match involves a logarithmic transformation of the square error terms and does not lend itself to a closed form analytical solution. We have used two algorithms for the calculation of the minimum representation transformation:

Numerical optimization - Partitioning of the search space using bounds on the volumes was implemented and combined with a random adaptive search for local minima. Hundreds of examples were studied using Monte Carlo techniques and the resulting transformations were examined and compared to mean squared error transformations. The minimum representation size results were stable and robust, particularly in the presence of added or missing data points.

Two-on-two transformations - It can be shown analytically that in one-dimension, a minimum representation transform always has two zero position error correspondences. In two dimensions, less than 1% of the optimal transformations found by simulation did not have two-on-two transforms, and for those cases the difference in the transformations was minor. We have therefore implemented a usually optimal transformation based on the two-on-two transformation. This approach dramatically decreases the complexity of the algorithm, reducing

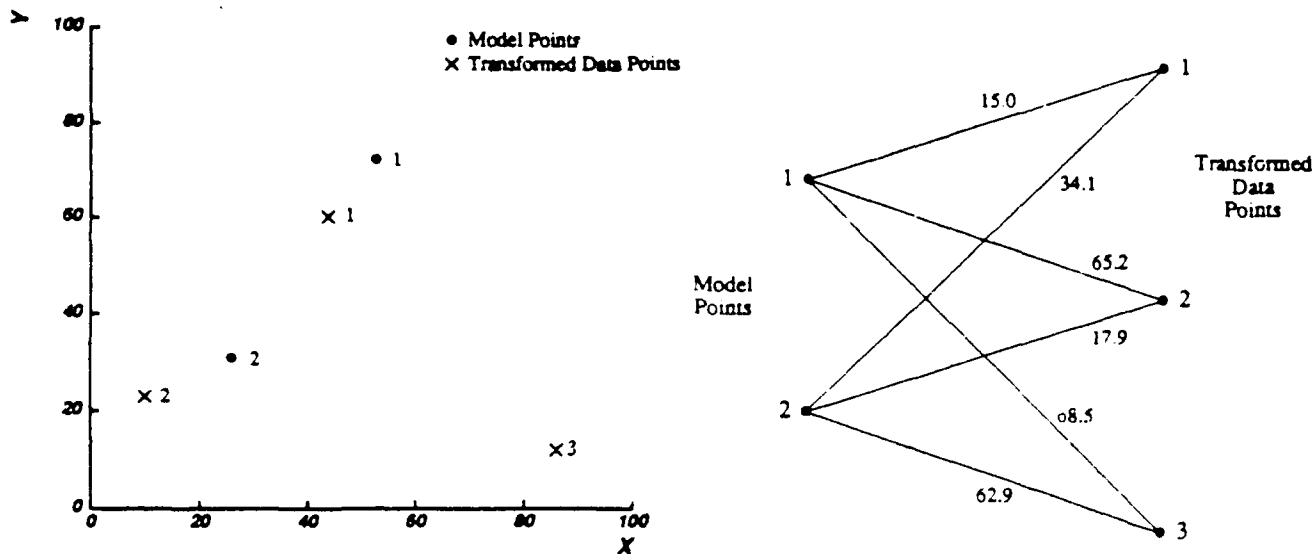


Figure 2 Two point sets and the bipartite graph of their possible correspondences.

a continuous parameter search in four dimensions to a discrete search over $(N^2 - N)/2$ points.

4.2 Correspondence

The correspondence problem of finding the match between subsets of data points and model points which minimizes the representation size is solved by converting it to an assignment problem in the following form. Based on the minimum representation size equation, each pair of model and data points has two alternative representations. As a modeled point, the pair may have a representation size, S_p , associated with the model and residuals. As an unmodeled point, the pair will contribute a fixed size S_a . Fig (2) shows a set of model points, a set of transformed data points, and a graph of their possible interpoint mappings. The transformation parameters are not optimal and were chosen for the purpose of illustration. The point numbers do not indicate correspondence. The graph of interpoint distances is a complete bipartite graph, and the optimal correspondence can be viewed as an optimal assignment of left nodes to right nodes which minimizes the representation size.

In order to calculate the optimal correspondence, we

1. Assign the pairwise representation size to each arc of the complete bipartite graph.

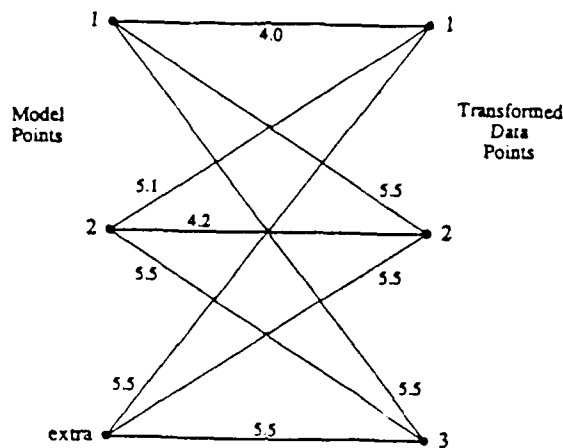


Figure 3 Expanded bipartite graph with representation sizes indicated as distance measures.

2. Replace those representation sizes which are larger than S_a by the value S_a
3. Let $N' = \max (M, N)$
4. If $M < N'$, add $N' - M$ 'extra' nodes to the set of model nodes. Connect each extra model node to every data node using N arcs, each with weight S_a .
5. If $N < N'$, add $N' - N$ 'extra' nodes to the set of data nodes. Connect each extra data node to every model node using M arcs, each with weight zero.

The resulting graph for Fig. (2) is shown in Fig. (3) with $S_a = 5.5$ bits. The optimal correspondence is now defined by choosing N' arcs such that (1) the sum of the arc weights is a minimum and (2) no two arcs share the same endpoint. A valid correspondence is indicated by a resulting arc weight which is less than S_a . All other arcs indicate that there is no correspondence between the two endpoints. The sum of the chosen arc weights is the representation size of the resulting match.

The assignment problem in a bipartite graph has been studied extensively [14], and a number of efficient algorithms exist. A straightforward solution would require evaluation of $N'!$ sets of arcs. Available algorithms typically are of order $O(N'^3)$ or $O(MN \min(M, N))$ [15]. The latter algorithm was implemented here.

4.3 Complexity

The complexity of the resulting algorithm may be summarized as follows:

1. Compute optimal two-on-two transformations - $O(M^2N^2)$,

2. Compute the graph of representation sizes - $O(MN)$,
3. Compute the optimal match using the assignment algorithm - $O(MN \min(M,N))$.

For large problems the computational complexity of the resulting algorithm is $O(M^3N^3 \min(M,N))$. While this algorithm still requires significant computation in its current form, on a typical size problem with $N = M = 30$, the computation is reduced relative to a brute force combinatorial algorithm by a factor of 1025. Many of the previous matching schemes have utilized heuristic techniques to reduce the computational complexity and did not optimize an objective measure of match quality. The algorithm described here produces usually optimal matches in polynomial time.

4.4 Improved Matching Efficiency

The performance of the basic matching algorithm can be improved using a number of algorithmic techniques and heuristics. The three methods summarized below utilize increasing assumptions about the characteristics of the data features.

1. **Precompute Representation Sizes:** The construction of the representation size graph requires the computation of MD representation sizes. Given a set of model points, it is possible to precompute all of the necessary representation sizes in a large x - y array. With such an array, the representation size calculation between the model point and any transformed data point is reduced to a single array access. Since a model is a collection of points, a number of separate arrays are required to represent all the possible representation sizes. The arrays are constant for a given model.
2. **Restrict Transform Space:** In most practical applications, there are fixed limits on the range of possible data point transformations. Those transforms which fall outside of this range can be ignored. In a typical vision application, the camera parameters are often fixed so that the scale of the data features is known within a few percent of their true value. With such scale, rotation, or translation restrictions, it is often not necessary to generate many of the candidate transforms, and the search space is correspondingly reduced.
3. **Approximate Method:** In the basic matching algorithm, we explore all possible transforms without screening the candidate matches based on error criteria. This approach has provided an accurate view of the performance of

the algorithm since it searches exhaustively over the candidates. In practice, one would like to reduce this search space based on prior screening of the errors. Such a fixed set of prescreened data points where only the most likely transformations and correspondences are explored, greatly reduces the search problem and adapts it well to practical situations.

5 EXPERIMENTAL RESULTS

The matching algorithm described in this report was tested on a variety of gray-level images with different degrees of complexity. Features extracted for matching are straight line segments and the vertices formed by the intersections and endpoints of such segments. The Popeye image processing system [16] was used to extract these edge-related figures by filtering, thinning, fitting of local line segments, logical reconnection, and simplification of the resulting line graph.

The line segments and their vertices are represented with a number of attached attributes. Each type of feature has a positional attribute. The position of a line segment is given by the center point of the line; while the position of a vertex is the point where two or more line segments intersect. In addition to the positional attribute, each segment also has a length attribute and a slope attribute. Vertex non-positional attributes include the number of line segments entering the vertex and the angle at which they enter.

Two examples of the feature extraction process are illustrated in Figs. (4) and (5). Fig. (4) shows a simple geometric shape with high contrast. The resulting edge-related features are clear and reliable as indicated by the dark lines and corner symbols in the figure. Fig. (5) shows a much more complex image which includes shading, highlights, and more subtle gray-tones. The resulting edge-related features are noisy and unreliable, and will often result in incomplete edge descriptions, or multiple vertices. Such complex images provide an important test of the minimum representation matching approach since they may contain a small percentage of repeatable features.

Fig. (6) shows an example of overlapping geometric shapes such as that in Fig. (4). These overlapping shapes provide a good test for the matching algorithm because they have occlusion among the objects. The contrast of the outer boundary of the shapes is still high, but the contrast among the objects is low

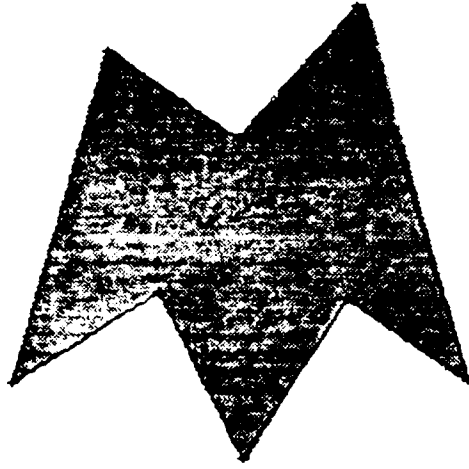


Figure 4 Feature extraction from a simple geometric shape with high contrast.

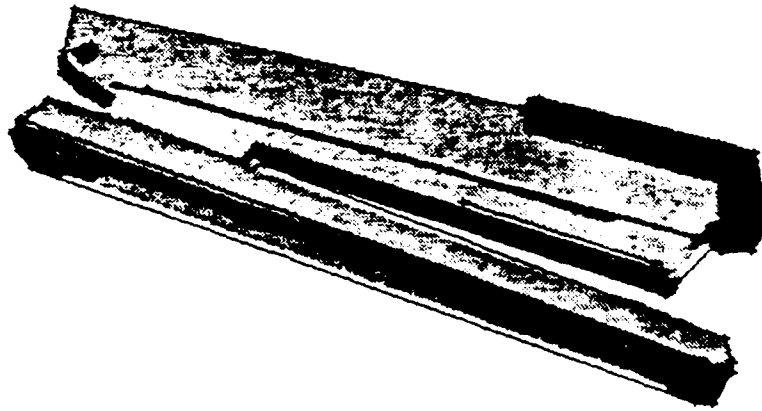


Figure 5 Features from a gray-level image of a three-dimensional object.

and in general do not provide edge-related features. In these experiments, each of the shapes was matched independently of the others, so that no constraints among the group of objects were used. For the experiments, the independent shapes were matched with high reliability.

The effect of employing non-positional attributes was studied for these geometric shapes and the results of a study of images with simulated distortions is shown in Fig. (7). In each case, a random subset of features were selected from

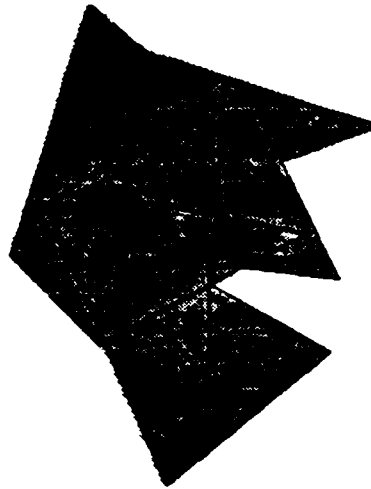


Figure 6 Example of minimal representation image matching with overlapping polygonal shapes. Models of the polygonal shapes were stored. Matching of each of the shapes to the gray level image was carried out independently.

Strategy	Vertex Attributes	Segment Attributes	% of Correct Matches
1	Position Only	Not Used	90.5
2	All	Not Used	99.5
3	Position Only	Position Only	96.0
4	All	All	100

Figure 7 Statistics for matching using several strategies with different incorporating different sets of attributes.

an image and a random set of synthetic features were added. Less than 50% of the features in all of these examples corresponded to the real image features. Four strategies were used on fifty examples of this type and the results are shown in Fig. (7). These results indicate that the algorithm is robust in spite of very large distortions of the data, and also that the addition of segment features, and the attributes for vertices and segments significantly improves the performance.

An example of a complex scene with an occluding object is shown in Fig.

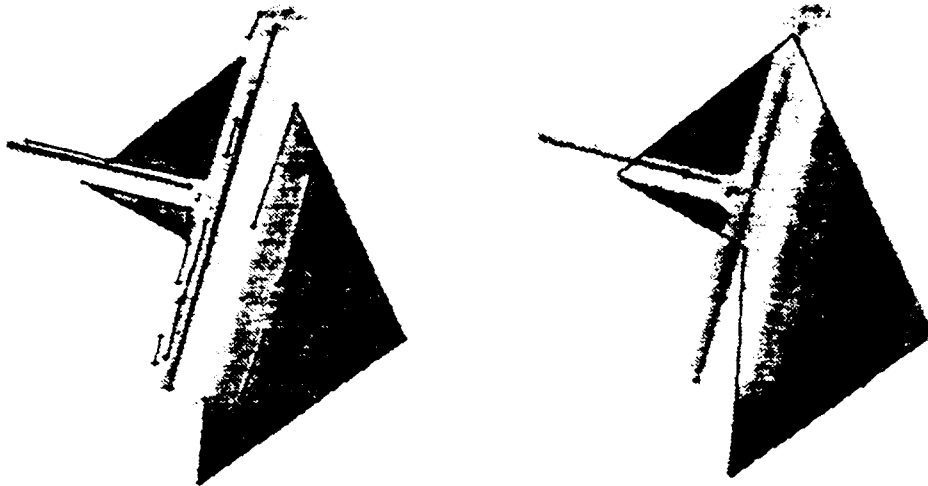


Figure 8 a. Example of image features for a noisy gray-level image of a polygonal shape with an occluding object. b. Correct matching of polygonal model to image features.

(8). The feature set derived from the original image is extremely noisy as shown in Fig. (8a). The correct match of the geometric model is shown in Fig. (8b).

Examples of matching to images of gray-level objects are shown in Figs. (9) and 10) for the example in Fig. (5). Fig. (5) shows the image and extracted features. Figure (9) shows the match of a model obtained from a slightly different angle of view. The resulting data image is quite noisy and varies significantly from the original model. The resulting match is still consistent with the model. Fig. (10) shows a match for an image of the object which is partially occluded. These noisy images typically had less than 40% consistent features as a basis for the match.

Experiments on RADC Images

The minimum representation matching algorithms described above have been applied to a variety of image test data made available from RADC. These test data included images of isolated aircraft and aerial images of airport scenes. In our experiments the isolated aircraft images were used as training images in order to derive models of aircraft shapes. The complex airport scenes were utilized for experiments in location and recognition of the model aircraft. The aerial images themselves were of highly variable quality due to imaging conditions, lighting

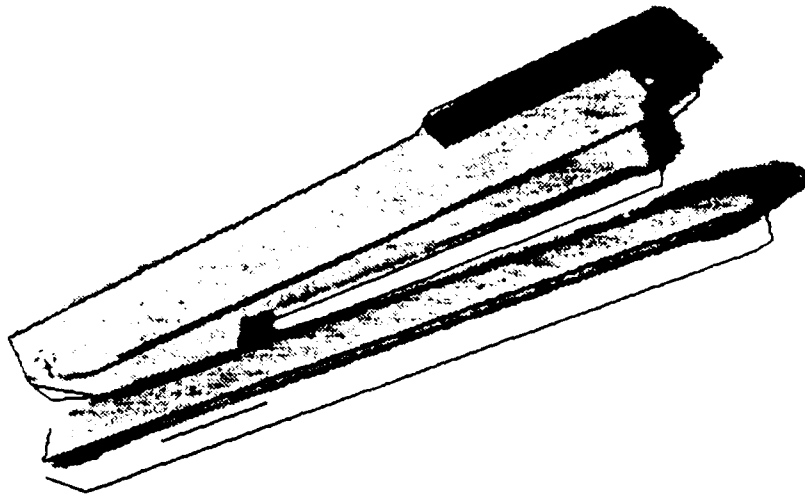


Figure 9 Matching of the gray-level image from Fig. (5) to a stored model obtained from the same object at a slightly different angle of view. Due to noise and distortion of the image, less than 30% of the features were consistent in this image, yet the algorithm was able to correctly match.

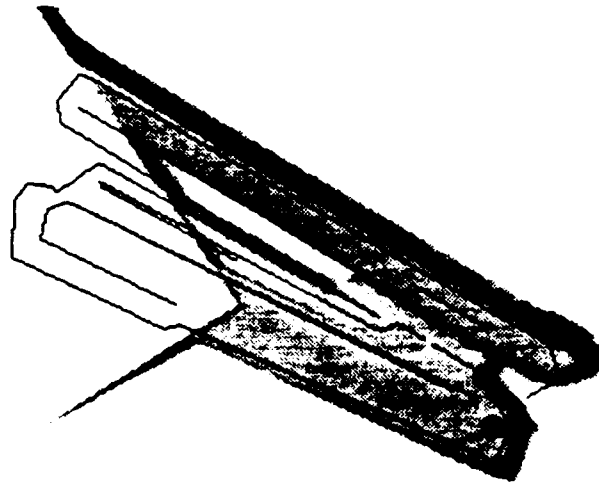


Figure 10 Matching of the stapler model to a gray-level image which is partially occluded.

conditions, and low resolution. We carried out preprocessing of these images using contrast enhancement techniques.

Extraction of the attributed graph representation from the image was accomplished using two different methods:

1. An edge extraction method similar to that used in the experiments described above. In this approach, continuity of contrasting edge elements in the gray level images was established and automatically simplified to produce a structured graph of nodes and edges.

2. The skeleton approach in which a morphological operator is used to extract a locus of central points in each gray level region of the image, and then these central points are connected into skeleton graphs consisting of nodes and edges.

In each case the resulting graph constitutes a data structure which is appropriate to the minimum representation matching methods described above. For many images the skeleton method was preferable due to the poor resolution of the aircraft in the aerial images, as well as the shape characteristics of the aircraft which consisted of configurations of thin black lines. Examples of matching based on the skeleton method are shown below.

Figure 11 shows an example of an isolated aircraft image used in these experiments. Figure 12 shows the skeleton-based attributed graph data extracted from this same image. The attributed graph structures contain additional associated attributes which include node positions, edge lengths, numbers of intersections, and vertex angles. The *minimum representation matching* utilizes all of these attributes.

Figure 13 shows a typical aerial photograph of an airport scene. In this scene there are several aircraft positioned on a runway, as well as a variety of other objects including buildings, foliage, and texture which is present on the runway. Variable lighting, noisy imaging conditions, and poor resolution contribute to the complexity of identification of the aircraft shapes within this image. Figure 14 shows a subimage which has been used for experiments on aircraft recognition. Figure 15 shows the skeletal data structure which is extracted from this airport image. The minimum representation matching algorithms were then utilized to identify candidate matches for the model data structure within this image. Figure 16 shows the resulting match of the model to the airplane positioned on the runway. It's clear from Figure 15 that there are a large number of possible node correspondences between the model and the data for this example. Based on the search over these possible correspondences, the position and orientation of the model is placed in such a way that it minimizes the overall representation size

as described earlier in this report. From intuitive interpretation of this image, the positioning of the model aircraft is correct within the tolerances of the imagery. Note that the minimum representation matching methods used are invariant to translation, rotation, and scale, and are tolerant to the distortion or inclusion of parts of the object.

Figure 17 shows an example of an airport subimage with more than one aircraft present. Figure 18 shows the extracted skeletal attributed graph representation for this image. Figure 19 shows the superposition of the model on the image in the position and orientation which corresponds to the minimum representation match. These results indicate a correct match to the first airplane in the set. The other airplanes in the image are also matched by the same model, but with higher representation size.

Often a correct, but distorted match resulted from use of the edge-based models described above. In this case the edge extraction from the low resolution image of the aircraft runway resulted in a noisy distribution of the edge points around the object. This noisy distribution of points in the blurred image resulted in the displacement of the model match and in the gradation in the accuracy of the resulting position and orientation of the model. The skeleton-based attributed graph extraction was overall a more reliable procedure for the experiments which we have carried out with this images.

These experiments have demonstrated the applicability of the minimum representation matching methods to the interpretation of aerial photographs of aircraft. The methods have been applied without modification to two different types of graph-based feature sets extracted from the image data. The results are characteristic of the reliability of the data points themselves, and the edge-based features provide less reliable matching due to the blur in the images associated with the resolution. The skeleton-based features are more reliable and are associated with the averaging affect which occurs in the application of the morphological operators to the gray level image.

6 CONCLUSIONS

This report has described a new approach to image matching which utilizes the minimum representation criterion as a means to obtain robust matching per-



Figure 11 Example of isolated aircraft image.



Figure 12 Skeleton-based attributed graph structure derived from the image in Figure 11.

formance even when image data is extremely noisy. The results are encouraging in that they demonstrate consistent performance on samples of real gray-level images. The computational complexity of the approach is polynomial, but still large for applications such as inspection and robot control. Additional simplifications and approximations have been suggested which might make the technique feasible in these domains, and parallel implementation may be required to make the computation time acceptable.



Figure 13 Example of an aerial photograph used in the matching experiments.



Figure 14 Subimage of the airport image in Figure 13.

The minimum representation approach to unsupervised decision-making is a general tool which has been employed in a number of different problems domains. The principle provides basic properties which seem to be useful in measuring and optimizing model structure as well as model parameters in a data interpretation framework. Such a minimum complexity or minimum entropy solution is appealing also from an intuitive point of view.



Figure 15 Skeleton attributed graph data structure derived from the airport image in Figure 13.



Figure 16 Resulting match of the model to the airplane positioned on the runway for the image shown in Figure 14.

REFERENCES

[1] J. Segen and A. C. Sanderson, Model Inference and Pattern Discovery by Minimal Representation Method, Technical Report CMU-RI-TR-82-2, Carnegie Mellon University, Robotics Institute, Pittsburgh, PA 15213, July 1981.

[2] J. Segen, A. C. Sanderson, and E. Richey, "Hierarchical Modeling and Classification of EEG Signals," *IEEE Transactions on Pattern Analysis and Machine*



Figure 17 Another subimage from the airport scene in Figure 13.



Figure 18 Skeleton-based attributed graph data structure for Figure 17.

Intelligence, vol. PAMI-2, No. 5, 405-415, September, 1980.

[3] J. Segen, *A Pattern Directed Approach to Signal Analysis*, PhD Thesis, Carnegie Mellon University, 1980.

[4] N. Foster, *Attributed Image Matching Using a Minimum Representation Size Criterion*, PhD Thesis, Carnegie Mellon University, 1987.

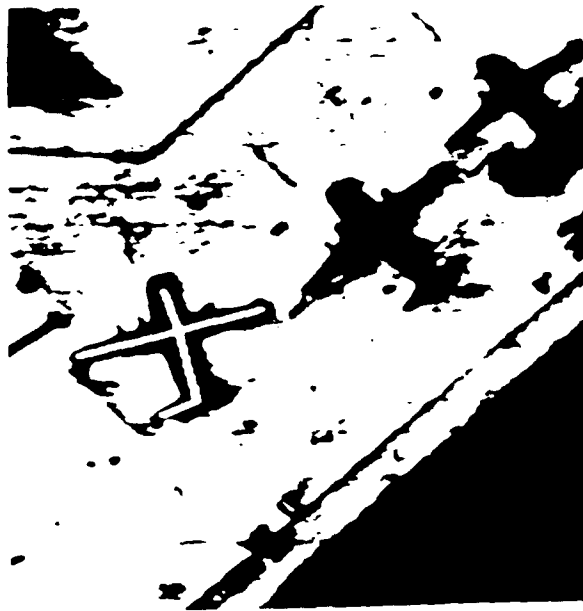


Figure 19 Superposition of the model on the image for a correct match for the minimum representation size. The other airplanes are matched successively at higher representation sizes.

[5] R. O. Duda and F. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, Inc., New York, 1973.

[6] H. S. Baird, *Model-Based Image Matching Using Location*. The MIT Press, Cambridge, Mass., 1985.

[7] M. A. Eshera and K. S. Fu. "A Graph Distance Measure for Image Analysis," *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-14(3):398-408, 1984.

[8] A. P. Ambler, H. G. Barrow, C. M. Brown, R. M. Burstall, and R. J. Popplestone, "A Versatile System for Computer-Controlled Assembly," *Artificial Intelligence*, 6(2):129-156, Summer, 1975.

[9] R. C. Bolles and R. A. Cain, "Recognizing and Locating Partially Visible Objects: the Local Feature-Focus Method," *The International Journal of Robotics Research*, 1(3):57-82, Fall 1982.

[10] L. G. Shapiro and R. M. Haralick, "Structural Descriptions and Inexact Matchings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(5):504-519, September 1981.

[11] N. Ayache and O. D. Faugeras, "HYPER: a New Approach for the Recognition and Positioning of Two-Dimensional Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1):44-54, January, 1986.

[12] J. Sklansky, "On the Hough Technique for Curve Detection," *IEEE Transactions on Computers*, C-27(10):923-926, October, 1978.

[13] L. Kitchen, "Relaxation Applied to Matching Quantitative Relational Structures," *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-10(2):96-101, February 1980.

[14] M. Gondran and M. Minoux, *Graphs and Algorithms*, John Wiley and Sons, Inc., New York, 1984.

[15] F. Bourgeois and J-C. Lassalle, "An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices," *Communications of the Association for Computing Machinery*, 14(12):802-806, December, 1971.

[16] R. Bracho, J. Schlag, and A. C. Sanderson, *Popeye: A Grey-level Vision System for Robotics Applications*, Technical Report CMU-RI-TR-83-6, Carnegie Mellon University, Robotics Institute, Pittsburgh, PA 15213, May, 1983.

AN APPROACH TO REPRESENTING SPATIAL INFORMATION USING GENETIC ALGORITHMS AND CLASSIFIER SYSTEMS

Michael M. Skolnick and David B. Jacobs

Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY 12180

1 INTRODUCTION

The attached paper reports on our research activities in using genetic algorithms to search the space of message forms provided by the Pebble_Pond algorithm. While our initial intent was to use the full-blown capabilities of the bucket-brigade algorithm to exploit the temporal structure provided by Pebble_Pond, we decided early on that the research issues involved in evolving chains of linked production rules were too substantial and would take us more into genetic algorithm/classifier system research as opposed to research on learning spatial structure. Thus, the attached paper reports on our restrictions to the structural variability of the message forms (provided by Pebble_Pond), which define the search space for the learning algorithm. In particular, we restrict our attention to simple pattern classes consisting of 2 or 3 points in each pattern instance (with many instances in each class allowing us to make the problem interesting). We were able to design a genetic algorithm which was capable of finding solutions for these simple problems. Issues in achieving these results, e.g., supporting speciation to combat premature convergence, are discussed in the paper.

In an attempt to consider less restrictive forms of messages we began to formulate a new genetic algorithm that would search the space of all possible embedded triangulation messages defined by the selections of 3 out of n points from each training image. While in the middle of this effort, we came to the conclusion that we needed to take a step back from moving on with the next logical step and instead reconsider some fundamental issues. One issue was that, while the restricted message forms which defined the search space worked well within the genetic algorithm, we were able to contrive examples of pattern classes which did not fall within the rubric of the message forms. In our initial restriction of the research away from a full blown bucket-brigade algorithm for forming arbitrary linkages between classifiers - which would have the potential to explore the space of various message forms themselves - we were certainly aware that we were picking off a subspace. In fact, we viewed the extension to the triangulation message forms as a way of exploring the restricted, but large, space of pattern classes characterized by angular relations between spatial locations. In addition, the counterexamples to this space all involved some commonalities in symmetry

that were not reflected in angularity. Thus, while these counterexamples were not unexpected, they brought back to our awareness the larger problem of finding ways of exploring the variety of message forms themselves.

At the same time, we also became aware of another fundamental issue in the way our fitness function guided the behavior of the genetic algorithm. Basically, the algorithm would allow parameterizations of various message forms to survive in the population, providing they were able to make some contribution towards correctly identifying the pattern classes. If a particular parameterization were capable of distinguishing all pattern instances correctly, then it would eventually spread through the population and dominate. Likewise, if a particular parameterization got many but not all of the classifications correct, the genetic algorithm could be tuned (once we learned a bit more about "speciation") to allow such parameterizations to survive. On the other hand, we were able to contrive examples where two parameterizations would contribute not at all to correct classifications, but a conjunction of the two would in fact be the answer. (These examples seem to correspond to what are called "maximally deceptive landscapes" in genetic algorithm research and, within the framework of pattern recognition, correspond to cases where all partial matches are incorrect, while the complete match is the answer.) The issue here is related to the one discussed in the above paragraph: how can the system automatically find non-summing (or non-linear) combinations of message forms when the set of patterns can not adequately be captured by the space represented either by individual message forms or simple additive combinations thereof. This of course is one of the fundamental issues in the realization of emergent phenomena and is an issue that appears in the biological literature in the framework of the debate about punctuated equilibrium versus gradualism.

As a continuation of this research we would propose to consider two basic tacts: 1) building upon the success in getting a working system: based on restricted message forms, we would consider adding on a system that "monitors" the behavior of the simpler message forms, seeking to determine when message forms might be combined into "higher-level" structures; this would correspond, within the framework of classifier systems as discussed by Holland, to the search for "arresting conditions" (or a priori constraints on the behavior of the adaptive system) that in effect operate *deus ex machina*, or 2) return to a consideration of the full-blown bucket-brigade system for dynamically combining primitive message forms into chains or groups of "higher-level" messages, that can then take on a unified role within the behavior of the system. Finally, as we continue to advance our understanding of the Pebble_Pond algorithm itself, we would expect that new measures and constraints on combinations of measures will arise and ultimately feed back into the research on spatial learning.

An Approach to Representing Spatial Information using Genetic Algorithms and Classifier Systems

**Michael M. Skolnick
David B. Jacobs**

**Department of Computer Science
Rensselaer Polytechnic Institute
Troy, N.Y. 12180**

Abstract

An approach towards representing spatial information within the context of genetic algorithms is presented. By translating positional information into a temporal sequence of bit string messages, it is possible to parametrize spatial information for use in genetic search. The transformation that produces the temporal sequence of messages is based upon a cellular automata simulation of wavefronts emanating from point sources. Message sets arriving early on report in parallel on all spatially proximate information, while latter message sets report on more disparate information. A genetic algorithm has been successfully applied to the search for classifiers that distinguish simple classes of spatial patterns. We consider how the various mechanisms of classifier systems, e.g., the bucket brigade algorithm and "triggering" conditions could be used to build a more robust system.

Introduction

The problem of representing spatial information on images via genetic algorithms and classifier systems (1) has proven to be quite difficult. Genetic algorithms are designed to effectively search a parametrized representation of space. That is, "genes" on an individual's "chromosome" correspond to the various classes of spatial measures, the values of the "alleles" correspond to the range of values within the space of measures and genetic operators are used to implicitly search for combinations of parameters (over all hyperplanes) that optimize the fitness function. With respect to spatial information the most immediate issue concerns the choice of an initial set of (hopefully robust) spatial measures. Also, there is the issue of searching for alternative measures if those initially chosen are inadequate. Even more fundamental is the extent to which the underlying representational framework of "chromosomal" bit strings and the associated genetic operators (especially cross-over) provide a good "match" with spatial structure.

One approach is to identify each gene with an image pixel where feature measurements at each pixel serve as alleles (2). One immediate problem with this approach is that feature measurements typically result in a rather sparse array of significant (above threshold) values, which translates into a sparse distribution of significant information spread about the individual "strings" in the population. In addition, translation, scale or rotational invariance is difficult to handle within such a framework. The most fundamental limitation of directly mapping pixels from euclidean space onto linear strings within genetic algorithms is that the mapping (and, in turn, genetic operators like cross-over) introduce discontinuities which disrupt the search for spatial structure. These issues have given rise - both within the area of genetic algorithms and computer vision in general - to approaches which are based upon defining a search space of parametrized spatial measures that capture the information within the pixel arrays. One GA effort in this direction has involved the search for optimal parametrized linear transforms that map one image array into another (3). This work, while of interest

from the perspective of estimating mappings between pixel arrays, is less relevant to the general problem of representing spatial information (unless it can be shown that the general problem can be cast in terms of image transformations).

Gillies [4] work stands out as an application of genetic algorithms involving the representation of spatial information. A learning system is developed that is able to distinguish between classes of input images provided as a training set. The learning system parametrizes the spatial structure through a class of image processing transformations and measures known as mathematical morphology [5]. The chromosomes used in this work were based on a morphological program "form" (or schema) that involve three broad classes of parameters, which represent the following spatial structure: 1. shapes that capture the structure of the background of the images, 2. shapes that capture the structure of the foreground of the images and 3. vectors that spatially relate the measures from foreground and background. Thus, there is a strong emphasis in Gillie's work on considering the search through measures of both shape and spatial position. In our work, we have focused exclusively on the issue of spatial position. Note that this does not preclude expressing shape information by combining primitive shape measures spatially. In fact, the tension between the exploration of robust shape measures versus their decomposition into spatial relations between more primitive shape measures is fundamental.

An algorithm has been developed, called Pebble_Pond [6], which can be thought of as transforming spatial structure into temporal structure. The algorithm is based upon a cellular automata simulation of wavefronts emanating from point sources. At each wave iteration within Pebble_Pond, filters and measures on the cellular array state space are taken, producing a set of messages about all significant "events" occurring at the iteration. Various measures and state space events have been shown to compute the following set of spatial structures: all order Gabriel graphs, Voronoi tessellations and nearest neighbor graphs [7], the point pairwise probability (or co-variance) distribution [5], detectors of approximately co-circular, co-linear or co-convex points sets, and new alternative measures of spatial structure. Based upon the diversity of the spatial measures provided by the cellular state space computed by Pebble_Pond, we feel that the state space might be an appropriate vehicle by which alternative spatial measures might be searched via a learning algorithm. In addition, at each iteration, i , measures on the changing wave configuration space are reporting on spatial structure that is separated by a distance $d(i)$. In other words, messages arising out of Pebble_Pond at time step i are reporting on all spatial structure a distance $d(i)$ apart and at latter time steps report on more distant spatial relationships. The result is a temporal ordering of messages coming out of Pebble_Pond that directly reflects a spatial ordering. With respect to learning algorithms, we feel that this transformation of spatial into temporal structure might provide an appropriate vehicle for the mechanisms of classifier systems, especially triggering conditions and the bucket brigade algorithm [1].

The next section will describe Pebble_Pond, with two purposes in mind: to define the set of messages that are used as input to the genetic algorithm and classifier system presented in this paper and to give some sense for the complete space of spatial structure that might be obtainable from Pebble_Pond. The third section will report on a genetic algorithm for learning to distinguish classes of spatial patterns based upon a specific class of input messages from Pebble_Pond.

The input: Pebble_Pond algorithm

Pebble_Pond, can be visualized in terms of the waves emanating from pebbles tossed into a pond of water. The wave propagation process differs from what happens in nature in that wave fronts - and the information they carry - are permitted to pass through each other without interference. The non-destructive intermingling of wave fronts permits the computation of non-planar graph structures. A significant aspect of the algorithm is its use of cellular automata transformations based on mathematical morphology [8,9,10]. These transformations provide a method for decomposing digital approximations of disks into sequences of local cellular neighborhood transformations. This is how: at each iteration within Pebble_Pond, the wave fronts of

increasing radii are generated. In addition, morphological filters permit the extraction - out of the complexity of wave space configurations that arise over time - of the information required for computing particular spatial structures.

To give the basic intuitions of Pebble_Pond and describe the messages input to the current genetic algorithm we will consider how Pebble_Pond can be used to compute the k -th order Gabriel Graph, $GG(k)$ [11; 12]. In order for an edge connecting two points to be in the $GG(k)$, the edge is considered to define the diameter of a circle and the resulting circle may circumscribe no more than k points. The first issue that needs to be addressed concerns how to translate the geometric criterion for constructing an edge in $GG(k)$ into an equivalent criterion involving wave phenomena. The basic observation exploited by the algorithm is the following: if circular waves are permitted to radiate from all points, then the waves of those points within the circle will reach the midpoint of the diameter before the wave fronts of the pair of points being evaluated as a Gabriel edge. Figure 1 illustrates this with respect to the algorithm determining that A-B and C-D form Gabriel edges. Note that for potential edge C-D there are three wave fronts that will have passed through the midpoint (indicated by the two darker spots) before the wave fronts from C and D meet; likewise, there are 5 such wavefronts passing through before those from A and B meet.

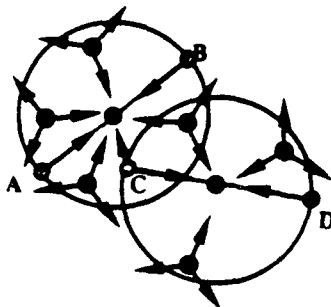


Figure 1: $GG(k)$ constraint - wavefronts within circles will hit midpoint diameters before circumference wavefronts.

If we consider that the algorithm must propagate the wave fronts of all possible point pairs while checking potential edge midpoints - all this in parallel - then it becomes clear that the algorithm is dealing with the manipulation of a large and highly complex set of cellular state space configurations. Thus, the main task of the algorithm for computing the $GG(k)$ is to spread the wave fronts in parallel, counting the number of wave fronts that have passed through all edge midpoints detected at the current time step. At the same time, it must do this without having any interference between the waves and without getting confused by the multiplicity of configurations. The details and issues on how this can be accomplished is described elsewhere [6] (Note that the Pebble_Pond algorithm is being implemented on an AIS-5000 linear array processor [13].)

Thus, significant information for the computation of $GG(k)$ resides at the midpoints of all edges of the complete graph. At each iteration i , Pebble_Pond detects all midpoints of edges some distance, $d(i)$, apart and then records the number of wave fronts that have previously passed through each midpoint. This information forms the structure of the input messages to the genetic algorithm /classifier system described in the next section. Thus, the three relevant message fields for $GG(k)$ are: the number of wavefronts - identified by "#-wave" - and the identity of the point source nodes - identified by "i" and "j" - of the potential Gabriel Edge. To this is added the following information fields (all of which can be made available by Pebble_Pond): "t" - the time step during which the wave fronts met (which is equivalent to one half the edge length), "x" and "y" - the x and y coordinates of the detected midpoint, and "theta" - the angle formed by the edge with respect to a

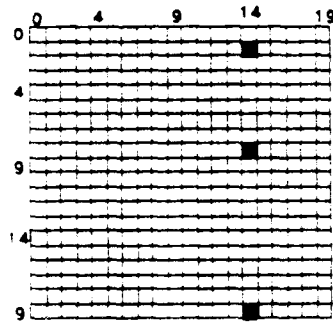
vertical axis. The training set of point patterns is divided into two classes and the genetic algorithm must learn how to distinguish them. Thus, for each image in the training classes, Pebble_Pond computes a set of messages in the order in which the wave fronts meet, where each message is of the following form:

```
t i j x y #-wave theta in-class
```

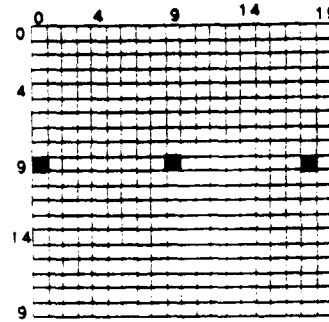
where

t is the time step at which the wave fronts of the Gabriel edge meet
i and j are the identity of the (Gabriel) nodes or point sources
(assigned in line scan image order)
and y are the (Gabriel) edge midpoints
#-wave is the number of wavefronts that previously passed through edge midpoint
theta is the angle (between 0 and 180 degrees) formed by edge and vertical axis
in-class indicates which of the two pattern classes to which the image belongs.

Consider the following two images both consisting of three points but representing different pattern classes (corresponding to vertically vs horizontally oriented point sets):



in-class = 1



in-class = 0

The messages, in order of arrival to the learning system, from the vertical point set are:

```
t:3 i:0 j:1 x:14 y:4 #-wave:0 theta:0 in-class:1
t:5 i:1 j:2 x:14 y:13 #-wave:0 theta:0 in-class:1
t:9 i:0 j:2 x:14 y:10 #-wave:1 theta:0 in-class:1
```

and from the horizontal point set are:

```
t:4 i:1 j:2 x:13 y:9 #-wave:0 theta:90 in-class:0
t:4 i:0 j:1 x:4 y:9 #-wave:0 theta:90 in-class:0
t:9 i:0 j:2 x:9 y:9 #-wave:1 theta:90 in-class:0
```

How the current genetic algorithm processes these message forms in order to distinguish the pattern classes will be described in the next section. The intent of initially restricting the learning algorithm to these message forms is to include an initial set of different basic types of spatial measures. For example, the time field, t , provides basic temporal information as does the order in which the messages arrive. In addition, when there are more points in the input images, $\#$ -waves will reflect some temporal information since midpoints of edges where the $\#$ -waves is large tend to be detected in latter stages of Pebble_Pond (unless there is something "special" about the point distribution). The four fields that encode wave identities (i and j) and midpoint coordinates (x and y) reflect more the absolute position of the point sources. Finally, the $\#$ -wave and theta fields are intended to reflect more the relative spatial positions of the images, where theta is rotational dependent and $\#$ -waves is rotational independent. The measures both enjoy some degree of independence and dependence and our idea is to use the genetic algorithm to explore different pattern classes with respect to understanding these measures. These measures seemed like a reasonable starting point from which to begin the exploration of genetic algorithms and classifier systems.

Before turning to the genetic algorithm that uses these messages as input, we conclude this section by describing some other measures provided by Pebble_Pond that may be significant. Besides functioning as additional dimensions to be added to the search space, these measures might function as "triggering conditions" (1) within a more general classifier system. For example, consider how Pebble_Pond can be used to compute the morphological co-variance distribution of point pairs (5). The distribution measures the probability of encountering pairs of points separated by all possible shift vectors in the image, i.e., the value of the probability density function at polar coordinates r and θ indicates the probability of encountering point pairs that are separated by a vector (r, θ) . The distribution is useful in general texture analysis and in applications involving the properties of materials (9; 14). Assume that, at each iteration of Pebble_Pond, the number of detected Gabriel edge midpoints is saved. The histogram of the number of waves meeting at each time step is an estimate of an orientation independent measure of pairwise co-variance, i.e., just with respect to r . This is obvious when we consider that all wave fronts from points the same distance apart will meet simultaneously. In addition, the full co-variance statistic - with orientation, θ , included - can be obtained via further processing of the cellular state space (6). In terms of triggering conditions, anytime a large (above "threshold") number of meeting events occur at a given time step, then that indicates potential interesting events. Anytime that there is some regularity in the time course of large numbers of meeting events, then there is evidence of textural information, e.g., consider a regular lattice (or texture) of points in which case there will be large numbers of meeting waves occurring over multiple time intervals based on the interval between neighboring lattice points. We are trying to suggest that there is the opportunity within the format of Pebble_Pond to define productions that are defined a priori and look for patterns of messages over time that reflect spatial structure.

A final example of another spatial structure provided by Pebble_Pond which should be useful in the development of a more robust classifier system is illustrated in Figure 2.

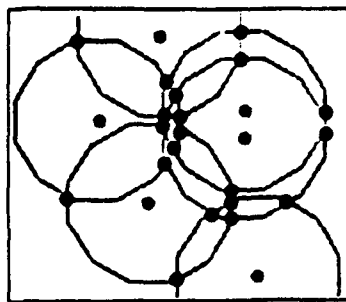


Figure 2: Detected wave crossings cluster at centers of "near" co-circularity

Basically, by defining appropriate morphological filters it is possible to detect all crossings of wave fronts at a given time step (6). Wave crossing events are significant, since any wave crossing represents a potential co-circularity of two point sources (see Figure 3). If at each time step, Pebble_Pond measures any clusterings of detected wave crossing, then in effect it is measuring how close point configurations can be approximated by a circle (whose radius is a function of the current time step). Information on near co-circularities - where they occur, when and if they repeat - is also fundamental spatial information that could be "looked-for" as potential triggering conditions. Relating such co-circular events across time (or equivalently, across different scale circles) may prove useful in the segmentation of important spot groupings. In addition to co-circularities, it is possible within Pebble_Pond to detect nearly co-linear point configurations and perhaps larger classes of (convex or non-convex) configurations. Within the context of learning algorithms one would expect the less specific measurements involving co-circularity and co-linearity to play a more fundamental role.

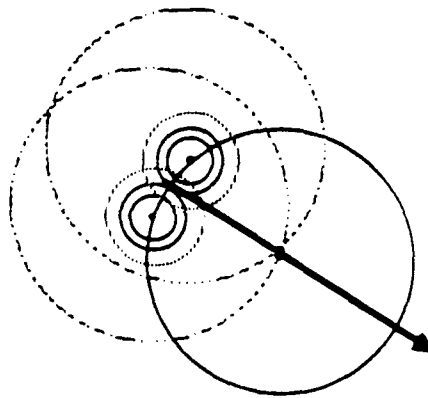


Figure 3: all crossing wave events represent potential pairwise co-circularities

The diversity of spatial structures produced by Pebble_Pond is a result of the measures and filters that can be applied to the evolving wave front configurations. Various filters can be chosen to be applied to the cellular automata state space at various points in the algorithm, resulting in different spatial structures or measurements. With respect to broad issues in learning algorithms, these measures may be chosen as initial input channels into a learning algorithm; it may also be possible to define learning algorithms which effectively search the "space" provided by Pebble_Pond to define new primitive or derived measures. Finally, specific patterns of measurements (e.g., indicating near co-circularities or texture information) might be provided a priori to a learning system as a way of implementing "triggering" conditions. We now leave these broad considerations and turn to our initial experiments using the message forms defined above.

Genetic algorithm for processing Pebble_Pond input

Given n points in a training image, there will be $\binom{n}{2}$ messages produced where each message has the following form (and number of bits as indicated in parentheses):

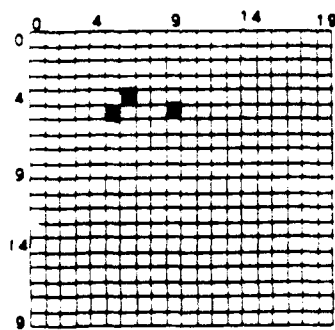
t	i	j	x	y	#-wave	theta	class
(7)	(4)	(4)	(7)	(7)	(4)	(8)	(1)

where all fields have the meaning described in the previous section and the class bit indicates which of two classes of images produced the message. Our broad goal is to build a classifier system that will search for those message forms that characterize the training sets. Obviously, since the number of messages grows at $O(n^2)$ the system should not require $O(n^2)$ classifiers, one for each possible message. Further, the eventual goal is to build up a hierarchy of classifiers making predictions and "higher-level" hypotheses over time. At the same time, there is a basic "locality" effect in the sense that messages, which arrive from different images at about the same time or within approximately the same order, should be primary candidates for discriminating the training sets. Thus, we envisioned the possibility of starting off with some large number of classifiers that would organize themselves into specialist "species" in the sense that each group would pay attention to spatial information at some specific time scale. While suggestive, this idea is way beyond what one would consider an initial step in the research - one must crawl before one walks. Thus, we decided to

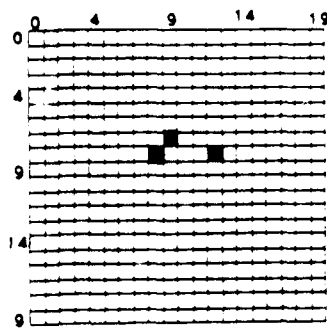
start our investigations by making the simplifying assumption that all $\binom{n}{2}$ messages from a training image,

would go into $\binom{n}{2}$ independent populations of classifiers attempting to categorize them, where the i -th population of classifiers is exposed to the i -th message produced by Pebble_Pond on each training image. If there are 6 training images (3 in the class and 3 not in the class) then population i would receive 6 messages from which to base its adaptation. As a simple example consider the following 2 classes consisting each of three images (and their point sets):

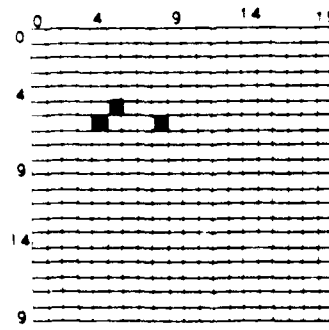
Images and messages from class 1



class1_1 = { (5,5) (9,5) (8,4) }



class1_2 = { (8,8) (12,8) (9,7) }

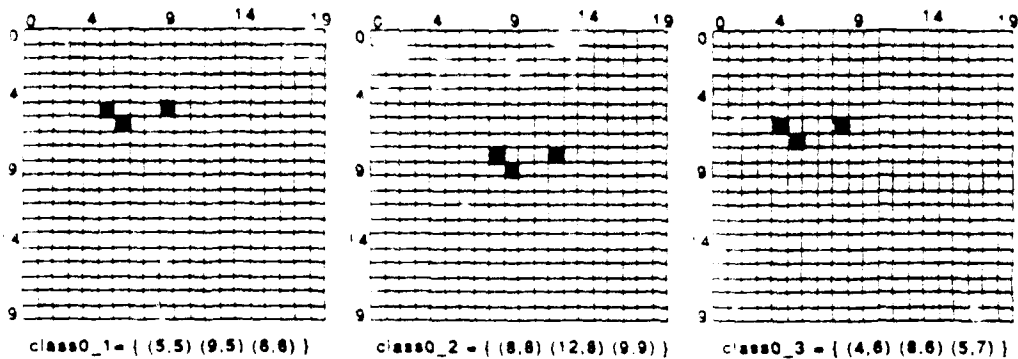


class1_3 = { (4,6) (8,6) (5,5) }

messages from each image in class 1 in order of arrival

class1_1							class1_2							class1_3									
<u>t</u>	<u>i</u>	<u>j</u>	<u>x</u>	<u>y</u>	<u>#-w</u>	<u>theta</u>	<u>class</u>	<u>t</u>	<u>i</u>	<u>j</u>	<u>x</u>	<u>y</u>	<u>#-w</u>	<u>theta</u>	<u>class</u>	<u>t</u>	<u>i</u>	<u>j</u>	<u>x</u>	<u>y</u>	<u>#-w</u>	<u>theta</u>	<u>class</u>
0	0	1	5	4	0	45	1	0	0	1	8	7	0	45	1	0	0	1	4	5	0	45	1
1	0	2	7	4	0	108	1	1	0	2	10	7	0	108	1	1	0	2	6	5	0	108	1
2	1	2	7	5	1	90	1	2	1	2	10	8	1	90	1	2	1	2	6	6	1	90	1

Images and messages from class 0



messages from each image in class 0 in order of arrival

class0_1							class0_2							class0_3									
<u>t</u>	<u>i</u>	<u>j</u>	<u>x</u>	<u>y</u>	<u>#-w</u>	<u>theta</u>	<u>class</u>	<u>t</u>	<u>i</u>	<u>j</u>	<u>x</u>	<u>y</u>	<u>#-w</u>	<u>theta</u>	<u>class</u>	<u>t</u>	<u>i</u>	<u>j</u>	<u>x</u>	<u>y</u>	<u>#-w</u>	<u>theta</u>	<u>class</u>
0	0	2	5	5	0	135	0	0	0	2	8	8	0	135	0	0	0	2	4	6	0	135	0
1	1	2	7	5	0	72	0	1	1	2	10	8	0	72	0	1	1	2	6	6	0	72	0
2	0	1	7	5	1	90	0	2	0	1	10	8	1	90	1	2	0	1	6	6	1	90	0

Note that in this simple example, it happens that the time the messages are generated corresponds to their order of generation. (In future systems, which will integrate information across different orders, rules focusing on a comparison of order and time of message generation could generate useful information.) To continue with the example, three sets of messages are sent to respective populations of classifiers (under separate genetic algorithms), where each set consists of all messages produced at that order in the output of Pebble_Pond. For example, the messages that are underlined are all input to the classifiers that are responsible for distinguishing the two image classes, where the distinction is based solely upon the first messages output by Pebble_Pond on each image. The messages (both in decimal and binary format) input to the classifier population for first order messages are:

Order 1 messages sent to classifier system/genetic algorithm

t	i	x	y	#-w	theta	class	image_id	t	i	x	y	#-w	theta	class					
0	0	2	5	5	0	135	0	class0_1	0000000	0000	0010	0000	101	0000	101	0000	10000	111	0
0	0	2	8	8	0	135	0	class0_2	0000000	0000	0010	0000	1000	0000	1000	0000	10000	111	0
0	0	2	4	6	0	135	0	class0_3	0000000	0000	0010	0000	100	0000	110	0000	10000	111	0
0	0	1	5	4	0	45	1	class1_1	0000000	0000	0001	0000	101	0000	100	0000	0010	110	1
0	0	1	8	7	0	45	1	class1_2	0000000	0000	0001	0001	1000	0000	111	0000	0010	110	1
0	0	1	4	5	0	45	1	class1_3	0000000	0000	0001	0000	100	0000	101	0000	0010	110	1

The above 6 strings of 42 bits then form the messages to a randomly initialized classifier system defined over the alphabet defined by {0, 1, #}⁴². To start our experiments, we used a fitness function with two components: The first component is a step function in which the classifier is given 42 points (the length of the bit string) for every correct classification of a given message and is penalized 42 points for each incorrect classification. A correct classification occurs when there is a match between the classifier and the message and the last bit (the bit which determines the class) of the classifier and message match. In the above case, if a classifier matched all 3 out of the 3 input strings in the class it is attempting to predict and did not falsely predict class inclusion for strings in other classes it would receive a maximum of 3 * 42 points towards its fitness. The second component makes the function more continuous by adding a bias towards matching individual bits in the message strings. To accomplish this we used the average number of bits for which the classifier matches a message. For example, if the classifier matched 15 bits of one message and 21 bits on another message, the result would be 18. The final fitness function is the addition of these two components. The genetic algorithm code was based upon some modifications to the CFS-C system [15] and we generally used most of the default parameters.

After running the genetic algorithm until it stabilized at maximum fitness classifiers - based on the system receiving the order 1 messages given above - the following classifier (placed beneath the input message strings) was one of those found with maximum fitness (of 166 = 3*42+40):

Order 1 messages sent to classifier system/genetic algorithm

t	i	x	y	#-w	theta	class				
0000000	0000	0010	0000	101	0000	101	0000	10000	111	0
0000000	0000	0010	0001	0000	1000	0000	0000	10000	111	0
0000000	0000	0010	0000	100	0000	110	0000	10000	111	0
0000000	0000	0001	0000	101	0000	100	0000	0010	110	1
0000000	0000	0001	0001	1000	0000	111	0000	0010	110	1
0000000	0000	0001	0000	100	0000	101	0000	0010	110	1

0000##0#0#00#0#####0#####0#001##|#00##|0#011#|1

By the way in which the training set is contrived - namely, that the difference between the classes resides in the angles formed by two of the edges of the triangles (with respect to a vertical axis) - the expectation is that significant bits will be found in the theta part of the classifier. For the optimal classifier above, the 4-th bit from the right ("1") turns out to be a significant bit. Note that there is no pressure in the fitness function towards #'s, in cases where a specific 1 or 0 either doesn't distinguish between the classes or does not interfere with the correct classification by the classifier. Other optimal classifiers of the population focused on other

significant bits of theta, e.g., the third bit from the left end point of theta. In addition, since the genetic algorithm is opportunistic, some of the optimal classifiers honed in on other (non-theta) differences in the inputs, e.g., bits in the values of j . This is to be expected in a situation where such a small sample of messages are guiding the learning.

Our next experiments involved more training examples so that the probability of random significant differences decreased. In addition, we sought to create training sets where a single critical distinction is insufficient to distinguish the classes, with the intention of having the genetic algorithm support two sub-optimal classifiers. It was relatively easy to control for random differences but, while in early generations the genetic algorithm brought out two suboptimal classifiers, any differences in fitness between the two eventually resulted in dominance by the better classifier. This was a result of the way in which parents are chosen and the method for determining which classifiers get replaced. Basically, parents are chosen according to the normalized probability based on each classifier's fitness. Replaceable classifiers are chosen with the probability of $1/\text{fitness}$. Unfortunately if left like this, it has the effect of quickly becoming a homogeneous population. While this is not harmful when the system needs to find one significant differentiating bit, it will cause serious problems if a class is defined by two independent criteria. This means that if being in a class signifies having A OR B, then there are two independent search spaces that the genetic algorithm needs to develop. One way to solve this is to create a pool of replacement classifiers based on $1/\text{fitness}$ and then choose which classifier to replace by which one is most like the replacing classifier. This has the effect of dividing up the population into different search spaces thereby allowing speciation. (Note to the reviewer: in a final paper we would expect to discuss these issues more systematically with more examples and more on the integration across the current partitions via orderings.)

Conclusion

Pebble_Pond can be viewed as a transformation that maps spatial structure, represented as points in a cellular array, into temporal structure. Information is produced at each iteration, which is based on selected measures of the state space of wave configurations, where early iterations represent more spatially proximate structure and latter iterations represent more spatially dispersed structure. For the initial design of the learning system discussed in this paper we focused on the information arriving from midpoints of the edges of the complete graph ordered in time according to edge length. As each configuration of points is presented to the learning system, Pebble_Pond will produce over time a set of "messages" that result from the various measures and filters defined on evolving wave configurations. The initial experiments discussed in this paper have treated message groups over time independently, but for simple discriminations the classifier system undergoing adaptation via a genetic algorithm has been able to find appropriate solutions. Work will continue on increasing the difficulty of the training sets, especially in terms of examples involving more complex correlations between measures. Eventually, we see a system developing that will use the bucket brigade algorithm to search for correlations of Pebble_Pond measures over time. In addition, the a priori spatial structure provided by Pebble_Pond, e.g., $GG(k)$, $VT(k)$, near co-circularities, can serve to define an initial set of measures upon which to define a classifier hierarchy. The genetic recombination procedures automatically search for relevant combinations of a priori measures and the mechanisms associated with triggering conditions provide the capability of adding new measures. Thus, we see both a rich a priori structure upon which to base a learning algorithm and a rich space of measures from which to discover new relevant structures.

6. References

- [1] Holland, J. H., Holyoak, K. J., Nisbett, R. E. and Thagard, P.R., Induction: Processes of inference, learning, and discovery, MIT press, 1986.
- [2] Englander, A. C., "Machine learning of visual recognition using genetic algorithms," Proceedings of first international conference on Genetic Algorithms, at Carnegie-Mellon University, 1985, pp. 197-201.
- [3] Fitzpatrick, J. M., Grefenstette, J. J. and Van Gucht, D., "Image registration by genetic search," Proceedings of IEEE Southeast Conference, 1984, pp. 460-464.
- [4] Gillies, A., Machine learning procedures for generating image domain feature detectors, Unpublished doctoral dissertation, University of Michigan, Ann Arbor, 1985.
- [5] Serra, J., Image Analysis and Mathematical Morphology, Academic Press, London, 1982.
- [6] Skolnick, M. M., Kim, S. and O'Bara, R., "Morphological algorithms for computing non-planar point neighborhoods on cellular automata," IEEE Second International Conference on Computer Vision, 1988, pp.106-111.
- [7] Preparata, F. P. and Shamos, M. L., Computational Geometry: An Introduction, Springer-Verlag, 1985.
- [8] Haralick, R., Sternberg, S. and Zhuang, X., "Image analysis using mathematical morphology," IEEE Pattern Analysis and Machine Intelligence, 9(4), 1987.
- [9] Serra, J., Image Analysis and Mathematical Morphology, Academic Press, London, 1982.
- [10] Skolnick, M. M. "Introduction to special section on Mathematical Morphology," Computer Vision, Graphics and Image Processing, Vol. 35, Number 3, 1986, p. 281-282.
- [11] Gabriel, K. R. and Sokal, R. R., "A new statistical approach to geographic variation analysis," Syst. Zool., Vol. 18, 1969, pp.259 -270.
- [12] Kim, S. M., Skolnick, M. M. & Kee, Y. Y., "Parallel and sequential algorithms for computing a generalized Gabriel graph," International Symposium on Optimal Algorithms, Bulgaria, June 1986, pp. 135-146.
- [13] Schmitz, L. A. and Wilson, S. S., "The AIS-5000 parallel processor," IEEE PAMI, 10(3), May, 1988, pp.320-330.
- [14] Skolnick, M. M., Brechner, E. and Marineau, "Morphological algorithms for analysis of geological phase structure," SPIE conference on visual communications and image processing, 1988.
- [15] CFS-C: a package of domain independent subroutines for implementing classifier systems in arbitrary user defined environments, Rick Riolo, Logic of Computers Group, University of Michigan.



MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control, Communications and Intelligence (C³I) activities. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C³I systems. The areas of technical competence include communications, command and control, battle management information processing, surveillance sensors, intelligence data collection and handling, solid state sciences, electromagnetics, and propagation, and electronic reliability/maintainability and compatibility.