

NAG1-412
JFR

P.119

LANGLEY GRANT

IN-61-CR

85446

FINAL REPORT: NASA-87

GRANT # NAG 1-412/JFR

FUZZY AUTOMATA AND PATTERN MATCHING

PRINCIPAL INVESTIGATORS: C. B. SETZER
N. A. WARSI



MATHEMATICAL AND COMPUTER SCIENCES
ATLANTA UNIVERSITY, ATLANTA, GEORGIA 30314

OCTOBER 1, 1983 - SEPTEMBER 30, 1986

NASA LANGLEY RESEARCH CENTER
TECHNICAL MONITOR: MICHAEL GOODE

(NASA-CR-181166) FUZZY AUTOMATA AND PATTERN
MATCHING Final Report, 1 Oct. 1983 - 30 Sep.
1986 (Atlanta Univ.) 119 p Avail: NTIS
HC AC6/MF AC1 CSCL 09B

N87-27429

Unclas

G3/61 0085446

FINAL REPORT: NASA-87

GRANT # NAG 1-412/JFR

FUZZY AUTOMATA AND PATTERN MATCHING

PRINCIPAL INVESTIGATORS: C. B. SETZER
N. A. WARSI



MATHEMATICAL AND COMPUTER SCIENCES
ATLANTA UNIVERSITY, ATLANTA, GEORGIA 30314

OCTOBER 1, 1983 - SEPTEMBER 30, 1986

NASA LANGLEY RESEARCH CENTER
TECHNICAL MONITOR: MICHAEL GOODE

TABLE OF CONTENTS

	<u>Page</u>
A. SUMMARY OF WORK	
a. A Survey of Pattern Recognition: An Annotated Bibliography	1
b. Edge Detection and Image Processing Using a Gradient and Edge Following Method	1
c. Minimal Finite Automata From Finite Training Sets	2
B. APPENDICES (Details)	
I. A SURVEY OF PATTERN RECOGNITION: AN ANNOTATED BIBLIOGRAPHY	5
II. EDGE DETECTION AND IMAGE PROCESSING USING A GRADIENT AND EDGE FOLLOWING METHOD	50
III. MINIMAL FINITE AUTOMATA FROM FINITE TRAINING SETS	108

FINAL REPORT

1. Project Period:

October 1, 1983 to September 30, 1986

2. Personnel:

Principal Investigators:

C. Bennet Setzer (Oct. 1, 1983 to Sept. 30, 1985)
N. A. Warsi (Oct. 1, 1983 to Sept. 30, 1986)

Graduate Students:

I. O'Nour, P. Sparrow, F. Brown, Linda Adams and E. Placide

The first three have finished their graduate theses and the last two are in the process of doing so.

3. Summary of Completed Work

This report contains the work accomplished through three main projects. Two more projects started during this period will finish shortly. The corresponding report will be included in that of the second project.

(a) A Survey of Pattern Recognition: An Annotated Bibliography (See Appendix I)

The work consisted of a wide-ranging search for articles and books concerned with fuzzy automata and syntactic pattern recognition. Also, a number of survey articles on image processing and feature detection have been collected. Although the work is basically survey, it lays down the foundation of further work through its annotated bibliography. The complete work is included as Appendix I.

(b) Edge Detection and Image Processing Using a Gradient and Edge Following Method (See Appendix II)

Hough's algorithm illustrates one way in which knowledge about the image can be used to interpret the details of the image. Since the used syntactic methods are more local, the performance of standard local techniques are studied. To this end, an edge following algorithm based on comparison of the gradients at adjacent pixels is implemented. In general, the closest match is considered an extension of an edge element. It was found that in hand generated pictures, the algorithm worked well on following straight lines, but had great difficulty turning corners. The connectivity of polygons was not evident in the resulting global edge trace. This experimental work although poor in performance has helped other related efforts. If an edge is being followed stepwise, some history would be useful in picking a continuation. The goal would be to be able to continue across apparent gaps in edges, given that the algorithm was already following a "strong edge". For details see Appendix II.

(c) Minimal Finite Automata From Finite Training Sets

The basic result of this research is an algorithm which produces a minimal finite automaton recognizing a given finite set of strings. Minimality is in the number of states and compared among those automata that recognize the given string and no other up to a given length. One difficulty of the construction is that, in some cases, this minimal automaton is not unique for a given set of strings and a given maximum length. Numerous examples show that it correctly deduces the "correct" automaton for sets of strings drawn from a given regular set. Further,

these examples show that non-unicity of the minimal automaton disappears for large samples. However, the convergence of the algorithm in the limit has not been demonstrated.

This algorithm compares favorably with other inference algorithms. It seems to produce smaller automata than other methods (e.g. [1]). It produces a deterministic automaton directly comparing with [1]. More importantly, the algorithm produces an automaton with a rigorously described relationship to the original set of strings that does not depend on the algorithm itself. In general, the study of abstract objects is more tractable if they have invariant relationships among themselves. See Appendix III for details).

[1] Vernadat, F and R. Rives, "Regular Grammatical Inference by a Successor Method", 6th International Conference on Pattern Recognition, Munich, 1982.

APPENDIX 1. A Survey of Pattern Recognition: An Annotated
Bibliography

1.1 INTRODUCTION

In a paper on pattern recognition, one would like to make the first statement a definition of pattern recognition.

" Mathematically, pattern recognition is a classification problem. We wish to design a system such that a handwritten symbol will be recognized as an 'A', a 'P', etc. The handwritten characters are often ambiguous, and there will be misclassified characters. The major goal in designing a pattern recognition machine is to have a low probability of misclassification. " [Young 1974].

" Pattern Recognition may be regarded as a process in which classes of patterns are directly mapped on a set of discrete elements. " [Kohonen 1984].

" Neither I nor any of my colleagues have found a succinct definition of pattern recognition (PR) which does justice to our personal understanding. PR means different things to each of us. " [Batchelor 1978].

Batchelor states the purpose of PR research as follows:

" The purpose of PR research is to build machines that will perform tasks which are essentially ' human '. PR is motivated partially by our desire to free human beings from tedious, boring, repetitive, or dangerous work. "

The problem of pattern recognition usually denotes classification and/or description of a set of processes or events. The set of processes or events to be classified could be a set of physical objects or a set of more abstract ones such as mental states. The processes or events with some similar properties are grouped into a class. The total number of pattern classes in a particular problem is often determined by the particular application in mind. For example, consider the problem of English character recognition; we should have a

ORIGINAL PAGE IS
OF POOR QUALITY

problem of 26 classes. On the other hand, if we are interested in discriminating English characters from Arabic characters, we have only a two-class problem. In some problems, the exact number of classes may not be known initially, and it may have to be determined from the observation of many representative patterns. In this case, we would like to detect the possibility of having new classes of patterns as we observe more and more patterns. Human beings perform the task of pattern recognition in almost every instant of their daily lives.

We could also think of a pattern, according to one of the dictionary definitions, as a model, guide, or plan used in making things. This can be a physical entity or an abstraction. Thus almost anything which is within the reach of our senses can be chosen as a pattern: a character, a photograph, a speech sample, odors, tastes, an idea, etc.

A pattern class is a group of patterns with certain properties. In rock patterns, the pattern classes may be various rock types: acidic, basic, calcic, sedimentary, etc. The problem of pattern recognition is that of classifying a pattern into one of the pattern classes on the basis of certain measurements or observations (subjectively).

Conceptually, the simplest form of recognition is probably "template matching". The sentence describing an input pattern is matched against sentences representing each prototype or reference pattern. [Fujita, 1971]. Based on a selected "matching" or "similarity" criterion, the input pattern is classified in the same class as the prototype pattern which is the "best" to match

ORIGINAL PAGE IS
OF POOR QUALITY

the input. The structural information is not recovered. If a complete pattern description is required for recognition, a parsing or syntax analysis is necessary. Recently, the use of discriminant grammars has been proposed for the classification of syntactic patterns.

Surveying the many different mathematical techniques used to solve pattern recognition problems, one can group them into two general approaches. These are the statistical or decision-theoretic (sometimes called discriminant) approach and the syntactic (or structural) approach.

Let us consider an unclass pattern recognition problem. If each pattern is considered as a single entity, then a set of n characteristic measurements (features) can be used to represent each pattern under study. In such a case, each pattern is represented by an n -dimensional feature vector and the recognition of patterns can be accomplished by applying various techniques in discriminant analysis and statistical decision theory. [10, 12, 11]. Such an approach is often called decision-theoretic (or statistical) approach.

On the other hand, if the patterns are very complex or if the number of pattern classes m is very large (in fingerprint identification or scene analysis), the number of features n required for recognition could also become very large. In such problems, the statistical approach often becomes ineffective or computationally infeasible in providing a solution. To overcome this difficulty, one could use syntactic methods or a hybrid of syntactic/semantic combination.

1.2 STATISTICAL PATTERN RECOGNITION:

In the decision-theoretic approach, the classification is based on a set of selected features extracted from the input patterns. These features' measurements are supposed to be invariant or less sensitive with respect to the commonly encountered variations and distortions and also contain less redundancies. The first subproblem is what measurements should be taken from the input patterns. Usually, the decision of what to measure is rather subjective and also dependent on the practical situations (e.g., the availability of measurements, the cost of measurements, etc.). The criterion of feature selection or ordering is often based on either the importance of the features in characterizing the patterns or the contribution of the features to the performance of recognition (i.e., the accuracy of recognition).

The second subproblem in pattern recognition is the problem of classification (or making a decision on the class assignment to the input patterns) based on the measurements taken from the selected features. The template-matching approach may be interpreted as a special case of the decision-theoretic approach, where the templates are stored in terms of feature measurements and a special classification criterion (matching) is used for the classifier.

Statistical pattern recognition is one of the mathematical theories underlying the design and analysis of recognition machines. By properly incorporating an input device (e.g. a scanner) to the digital, analog, or hybrid computer system, a statistical recognition machine is formed. In addition to the

characters and biomedical waveforms. Statistical recognition methods have been successfully applied to patterns in weather prediction, photo-interpretation, multispectral crop classification, and many other areas. [Chen'73].

1.3 DISCRIMINANT FUNCTIONS AND FORMULATION OF PATTERNS

Let us consider patterns which are represented as vectors in the multidimensional Euclidean space R^n . One of the most fundamental approaches in pattern recognition is the classification of patterns by means of discriminant functions [Kohonen'84]. Assume that the representation vectors are grouped into a finite number of clusters (each of which corresponds to a particular class). How can one define (mathematically) the equations of those hypersurfaces which optimally separate all clusters from one another? The simplest alternative is to try the linear hyperplane (the lowest degree surface). If the linear separation fails, one may take polynomial forms of successively higher degree. One optimality criterion could be the distance of all points from the separating surfaces. One frequently used criterion is the sum of squares of Euclidean distances.

Pattern classification is a process that is related to decision making, detection theory, etc., and may be discussed in these settings [Kohonen'84].

1.4 REPRESENTATION OF PATTERNS:

We must distinguish between a pattern (e.g. a visual image) and its representation within a computer- a process during which a lot of information was destroyed (because pattern description causes destruction of information).

If the input to a pattern recognition machine is a set of N measurements (e.g. an N -dimensional vector), the output is the classification C . [Young'74]. Since C depends on the input vector X , the classification is written as:

$$C = d(X)$$

where $d(X)$ is the decision function.

The selection of measurements is very important in the design of these machines. Because the values of X are determined by the measurements, the measurement selection defines the the pattern space S_X .

The pattern recognition machine may be divided into two parts, a feature extractor and a classifier. The feature extractor reduces the dimensionality of the input vectors to the classifier. Hence feature extraction is a transformation

$$Y = T(X)$$

which transforms a pattern vector X in the pattern space S_X into a feature vector Y in a feature space S_Y . The classifier then classifies X based on Y . Because S_Y is of lower dimensionality than S_X , the transformation is singular and some information is lost. The feature extraction should reduce the dimensionality but at the same time maintain a high machine performance.

ORIGINAL PAGE IS
OF POOR QUALITY

A special case of feature extraction is feature selection. It selects as features a subset of the given measurements.

1.5 FEATURE EXTRACTION

" Feature selection (or extraction) is generally considered a process of mapping the original measurements into more effective features. If the mapping is linear, the mapping function is well defined and our task is simply to find the coefficients of the linear function so as to maximize or minimize the criterion. Therefore, if we have the proper criterion for evaluating the effectiveness of features, we can use the well developed techniques of linear algebra for simple criteria. or, in the case of a complex criterion, we can apply optimizing techniques to determine these mapping coefficients. Unfortunately, in many applications of pattern recognition, there are important features which are not linear functions of original measurements, but are highly nonlinear functions. Then, the basic problem is find a proper nonlinear mapping function for the given data. Since we do not have any general theory to generate mapping functions systematically and to find the optimum one, the selection of features becomes very much problem oriented. " Fukunaga 1972.

One process frequently used in data reduction is feature extraction. Patterns (ordered sets of values) in practice may have dimensionalities that are intolerably high. Therefore it is common to preprocess these values, by forming various functionals over selected subsets of the pattern elements [Kohonen'84]. Such functionals are then named features, and in many cases they contain important correlations or other relationships which comprise the intrinsic information. The feature may be chosen statistically or heuristically.

Heuristics derive its name from the famous exclamation of Archimedes: "Eureka" - 'I have found'. It studies creative activity [Pekelie'74].

Heuristics, according to Judea Pearl, 1984, are criteria.

ORIGINAL PAGE IS
OF POOR QUALITY

methods, or principles for deciding which among several alternative courses of action promises to be the most effective in order to achieve some goal. They represent compromises between two requirements: the need to make such criteria simple and, at the same time, the desire to see them discriminate correctly between good and bad choices.

In a pattern recognition problem, a number of pattern features can be extracted from a pattern, but only a few of them are often necessary to the recognition. To determine what variables (or attributes) are to be measured is the most important step towards designing an efficient recognition machine [Chen'73]. The task of determining the variables is called feature selection or extraction. Feature selection is usually preceded by preprocessing (data correction) of patterns, which refers to the processes of sorting, normalization, filtering, etc. of the input patterns. The desirable requirements of the features selected are:

- (1) they properly describe the pattern,
- (2) they are easy to process, and
- (3) they are invariant to translation and rotation of the pattern.

The selectivity in the choice and the use of pattern features seems to belong to the fundamentals of human recognition. [Pik]. In the statistical approach to pattern recognition, this natural procedure results in sequential methods.

Another approach for the reduction of data to be stored in a recognition is to perform a classification of primary data (e.g., pattern vectors) into representative subsets. These subsets could be structurally related ignoring the individual data points.

There are many instances where classification must and can be performed without a priori knowledge [Fukunaga 1972]. The clustering problem is not well defined unless the resulting classes of samples are required to exhibit certain properties. The choice of properties (or the definition of a cluster) is the fundamental issue in the clustering problem. Given a suitable definition of a cluster, it is possible to distinguish between good and bad classifications of samples.

The clustering problem may be set up, [Kohonen'84], as follows: assume that

$$A = \{ a_i : i = 1, 2, \dots, n \}$$

is a finite set of representations of items. This set has to be partitioned into disjoint subsets

$$A_j, j = 1, 2, \dots, k$$

such that with respect to this division, some functional describing the 'distance' between items attains an extremum value. This functional ought to describe the quality of groupings in the sense that the mutual distance between all a_i belonging to the same A_j are as small as possible while the distances between different A_j are large. The functional describing the grouping may contain, e.g., sums of some powers of the distances.

CHAPTER TWO

THE SYNTACTIC APPROACH TO PR

2.1 SYNTACTIC PATTERN RECOGNITION:

" In some recognition problems, the structural information that describes each pattern is important, and the recognition process includes not only the capability of assigning the pattern to a particular class (to classify it), but also the capacity to describe aspects of the pattern that make it ineligible for assigning to another class. " [Fu'82].

" The syntactic methods begin with the extraction of primitives from the patterns. With regard to line drawings, the primitives can be line segments defined, e.g., by regions of certain curvature, or segments between distinct points such as intersections or branchings. In principle, any subpattern in the original image, when suitably segmented and labelled, could also be defined as a primitive. " [Kohonen'84].

" The study of syntactic pattern recognition has intensified during the last few years. In order that more experience shall be gained with the approach it is important that it can be applied by researchers whose principal interest is the application rather than the syntactic pattern recognition method. Thus there is a need for a suite of programs that enables a researcher skilled in neither language theory nor computing to make progress with the method." [Blake'81].

A typical example of the class of recognition mentioned above is scene analysis (in particular picture recognition). Here, the patterns under consideration are usually quite complex and the number of features required is often very large. In this kind of problems, the syntactic approach to pattern recognition is more appropriate. This approach draws an analogy between the (tree-like) structure of patterns and the syntax (or grammar) of languages. Patterns are specified as being built up out of subpatterns in various ways of composition, just as phrases and sentences are built up by concatenation of words (and words by concatenating characters). The simplest subpatterns (pattern

ORIGINAL PAGE IS
OF POOR QUALITY

primitives) selected, should be much easier to recognize than the patterns themselves.

2.2 PRIMITIVE SELECTION AND PATTERN GRAMMARS

Though pattern primitives are the basic components of a pattern, they are not easy to recognize. For example, strokes are considered good primitives for script handwriting, and so are phonemes for continuous speech; however, neither strokes nor phonemes can easily be extracted by machine. A compromise between its use as a basic part of the patterns and its easiness for recognition is often required in the process of selecting pattern primitives. [Fu'82].

So far there is no general solution for the primitive selection problem. For line patterns or patterns described by boundaries of skeletons, line segments are often suggested as primitives. The information characterizing the primitives can be considered as their associated semantic information or as features used for primitive recognition. Through the structural description and the semantic specification of a pattern, the semantic information associated with its subpatterns or the pattern itself can then be determined.

After pattern primitives are selected, the next step is the construction of a grammar which will generate a language to describe the patterns under study. It is known that increased descriptive power of a language is paid for in terms of increased complexity of the syntax analysis system (recognizer or acceptor). The selection of a particular grammar for pattern description is affected by the primitives selected and by the

ORIGINAL PAGE IS
OF POOR QUALITY

tradeoff between the grammar's descriptive power and analysis
efficiency.

ORIGINAL PAGE IS
OF POOR QUALITY

Attributed grammars were first formulated by Knuth to assign semantics (or meanings) to context-free languages from the computational point of view. Each production rule of an attributed grammar consists of a syntactic rule (to specify language syntax) and a semantic rule (to add contextual semantic).

The following is a definition of attributed grammars according to the formalism of Knuth:

An attributed context-free grammar is a 4-tuple

$$G = (V_N, V_T, P, S)$$

where

V_N	set of nonterminals,
V_T	set of terminals,
$S \in V_N$	start symbol,

for each $X \in (V_N \cup V_T)$, there exists a finite set of attributes $A(X)$, each attribute α of $A(X)$ having a set, either finite or infinite, of possible values D_α ; and P is a set of productions each of which is divided into two parts: a syntactic rule and a semantic rule.

In practical applications, pattern classes can often be divided into groups, each group consisting of several pattern classes which are similar in structure but different in attributes. In such cases, it is appropriate to construct an attributed grammar for each group of pattern classes. The discrimination of within-group pattern classes is left to statistical classification on attributes.

3.1 THE COMBINED SYNTACTIC/STATISTICAL APPROACH

" Thus the controversy between geometric and structural approaches for problems of pattern recognition seems to me historically inevitable, but temporary. There are problems to which the geometric approach is ideally suited. Also, there are some well known problems which, though solvable by the geometric method, are more easily solvable by the structural approach. But any difficult problems require a combination of these approaches, and methods are gradually crystalizing to combine them: the structural approach is the means of construction of convenient space; the geometric is the partitioning in it "[Aizerman'69].

Combining syntactic and statistical pattern recognition approaches, has been advocated by several investigators in the past decade. [Tsai'90]. The motivation arises from the fact that neither the syntactic approach nor the statistical approach alone is adequate for some practical applications. The syntactic is weak in handling noisy patterns and numerical semantic information. Another characteristic drawback of structural methods is the arbitrariness in the selection of primitives (i.e., there exists an infinite number of ways for the construction of grammars for pictures). The statistical is incapable of describing complex pattern structures and subpattern relations. Since the advantages of one seem to be the drawbacks of the other, a hybrid model that incorporates the advantages of both is desirable in dealing with real applications. The use of attributed grammar was suggested by Tsai and Fu, 1990, as such a tool.

ORIGINAL PAGE IS
OF POOR QUALITY

authors have formulated a quite general and powerful scheme for pattern analysis which can be viewed as a combination of syntactic and statistical approaches to pattern recognition.

ORIGINAL PAGE IS
OF POOR QUALITY

attributed grammars. The semantics computation is performed simultaneously to obtain all required nonterminal (subpattern) attributes according to the semantic rules. It is possible, during the computation of semantics using the semantic rules, that some nonterminal (subpattern) attributes may not be obtainable by a mapping or computation from lower level terminal attributes. When that occurs, it is necessary to go back to the input pattern, to find out the subpattern corresponding to the nonterminal, and to perform the necessary subpattern attribute extractions as specified in the semantic rule. The authors would like to emphasize here that subpattern attributes cannot be extracted before syntax analysis and semantic computation because, without the guidance of syntax analysis, the system would not know which terminals (primitives) should be grouped into a nonterminal (subpattern). This is indeed an advantage of using attributed grammars because subpattern attribute extraction now becomes more effective with the guidance of syntax analysis (Tzeng 1981). Such an advantage is not obtainable by using the statistical approach alone. After this stage, the result is a parse of the pattern representation if the input is syntactically correct, together with its total attribute vector. The parser will reject any syntactically incorrect representation corresponding to a structurally erroneous pattern. The hierarchical syntactic and semantic description of the input pattern is also available now except its class assignment.

The final stage is the decisionmaking performed on the total attribute vector to classify the input pattern. By combining syntax analysis, semantic computation, and decisionmaking, the

ORIGINAL PAGE IS
OF POOR QUALITY

From the statistical classification point of view, the incorporation of syntax analysis into statistical decisionmaking has several advantages over the classical statistical approach:

- (1) utilization of structural information for pattern description,
- (2) effective extraction of subpattern attributes, and
- (3) description and generation of patterns.

On the other hand, from the syntactic point of view, the injection of primitive and subpattern attributes into the grammatical analysis offers the following advantages over the conventional syntactic methods:

- (1) flexibility in choosing primitives and subpatterns,
- (2) improvement of recognition accuracy,
- (3) capability of recognizing noisy patterns, and
- (4) reduction of grammatical complexity.

3.2.2 A PATTERN ANALYSIS SYSTEM USING ATTRIBUTED GRAMMARS

The block diagram of a syntactic pattern recognition system using attributed grammars is shown in Fig. 2. Given an input pattern for classification, after preprocessing, all necessary primitives and their attributes are extracted according to some prepecified procedures [Tsai'80]. No subpattern attributes are to be extracted at this stage because they are still unknown. The next step is to transform the primitive set into some structural representation (a string, a tree, etc.) by assigning symbols to primitives, selecting concatenating directions, or any other prepecified relations. The resulting representation is then analyzed syntactically by using the syntactic rules of the

CHAPTER FOUR

PRACTICAL APPLICATIONS OF PR

4.1 BIOMEDICAL APPLICATIONS

The medical applications of pattern recognition are quite numerous, and among the most interesting is electroencephalography.

4.1.1 ELECTROENCEPHALOGRAPHY

The electroencephalogram (EEG), is a complex electrical signal which reflects generalized brain activity. The EEG is utilized in the clinical assessment of many neurological and psychiatric disorders and offers promise for monitoring of patients undergoing anesthesia and operation [Sanderson'80]. Automated analysis and classification of EEG signals is hampered by the enormous amounts of data which are generated by detailed descriptions of the nonstationary waveforms.

In order to detect abnormalities, one needs to have a standard domain of the 'normal'. EEG have helped in dealing with epilepsy. The literature contains several systems for the automatic recognition of abnormalities associated with epilepsy. Virtually most of the systems developed have dealt with the detection of spikes and sharp waves [Thorne'81]. One of the most frequently used approaches has been to compare the amplitudes, derivatives and durations of EEG wave forms with respect to preset thresholds to indicate the presence of abnormalities.

4.1.1.2 BRAIN ELECTRICAL POTENTIALS

Few problems are more challenging than decoding the mass electrical activity of the human brain [Gevins'80]. Although this goal is distant, there are currently several useful applications of a correlative approach based on pattern recognition. These include investigation of the neural concomitants of higher cognitive functions, diagnostic screening, differential diagnosis and prognostic assessment of neurologic disorders, classification of the stages of ordinary disturbed sleep, and classification of the neuroelectric patterns associated with different types of psychotropic drugs. Since brain electrical potentials (BEP) are correlated with a variety of behavioral and clinical variables, especially tight experimental designs are necessary.

If features are not neurophysiologically interpretable, the utility of even successful classification is likely to be limited in most applications. Features should be expressed in a form which is insensitive to irrelevant anatomic and metabolic variables. Sample size determination is application-specific since intra- and interperson BEP variability is different for different problems [Gevins'80].

Most pattern classification algorithms have been applied to BEP's including decision functions, trainable classification networks, distance functions, syntactic methods, and hybrids of the preceding.

4.1.3 EXPERIMENTAL PSYCHOLOGY

Experimental psychologists have been interested in the perception of symmetry since the last century [Schaeffer '9*]. The interest has been mainly with the perception of bilateral symmetry. At first glance translational symmetry would appear to be the simplest of the symmetry operations, yet, it is bilateral symmetry that is detected more quickly and with greater accuracy than the other symmetries. A computational model of human symmetry detection was developed by Schaefer for detecting the presence of the various symmetries found in two dimensional shapes. This model utilizes search strategies and a comparison process for detection. The search strategies consist of the successive selection of feature pairs to compare for the presence or absence of symmetry. A basic search strategy was specifically designed to reduce the number of feature pair comparisons carried out by capitalizing on the differences, in the distances between symmetrically related features, that are found in the various symmetries. The results indicate a strong relationship between the behaviour of the model and that of human subjects on symmetry detection tasks.

4.1.4. EE APPLICATION TO CANCER

"Breast cancer is the leading cause of death in occidental women. According to statistics, the death rate is 22 per 100,000 female population in the United States for the past 40 years. Early detection increases the five-year survival rate to 90-95% in contrast to 40-50% where detection of breast cancer is delayed."

Silverberg '71.

According to Fong et. al. (1994), thermography is regarded as a potential screening tool for early detection of breast

cancer. 'but its value is still in doubt'. They conducted a study designed to critically evaluate the validity of thermographic images as an aid to the early detection of breast cancer.

The following is a brief description of the procedure they followed. A set of 24 elemental features was extracted from thermographic images and combined into 16 compound features. These features were used as input to the pattern recognition system using generalized inverse approach. From the result of the pattern recognition, it is found that thermography is not suitable for early breast cancer detection.

4.1.5 EE APPLICATION TO CAROTID WAVES

Methods of syntactic pattern recognition and knowledge based systems for signal understanding have recently been proposed for application in biomedical engineering [Arduino'92]. Among the application of such methods, a system for the interpretation of carotid pulse waves using a general waveform parsing system has been developed by Stockman, Kanal and Kyle.

The motivation for analyzing the carotid signal resides in the fact that carotid artery in the neck is the point in the arterial system close to the heart where external (non-invasive) sensing can be carried out. The carotid pulse wave exhibits wider structural variations than pulse waves from other body points. It is believed that more valuable information, related to age and cardiovascular disease can be gathered from the carotid pulse wave.

Though the work proposed in the Arduino paper follows the lines of Stockman, Kanal and Kyle, important innovations

ORIGINAL PAGE IS
OF POOR QUALITY

characterize the Arduino interpretation. One of these is that a functional approximation of the waveform of the search for important features on it is performed after an automatic extraction of the fundamental period. This allows one to make a 'best-synchronous' analysis with advantages already tested in systems developed for the automatic processing of electrocardiograms (ECG).

4.2.1 RE APPLICATION TO CHARACTER RECOGNITION

The problem of machine intelligence attracted scientists from the moment the first computer was born [Verschueren '84]. Especially the automatic recognition of data generated by humans (e.g. alphanumeric characters) was a favourite problem in those early days. Some big computer companies spent a great amount of money into this research. Later on the problem was abandoned a little because of the large amount of computing power needed. Today, with the enormous amount of information produced by men and machine, the problem of automatic recognition of both hand-oriented and machine printed characters is again up to date. Many companies try to find a solution to the amount of administration they must treat. Although many years have been spent on this research, no general solution exists. The methods developed "are all partially empirical and suited for one particular problem" [Verschueren '84]. One of the most difficult problems in this area is the recognition of handwritten characters.

1.2.2 CURSIVE SCRIPT

ORIGINAL PAGE IS
OF POOR QUALITY

"The central problem of cursive script interpretation is that of determining the identity of connected spatial symbols that correspond to words" [Bosinovic '84].

The problem of cursive script differs from optical character recognition which predominantly deals with discrete letters, and is akin to speech recognition dealing with continuous strings of phonemes. Research on word-level recognition of off-line cursive script started almost twenty years ago and has been revisited only occasionally [Amin '84]. The recognition method depends on the type the characters to be recognized. For the Arabic language, e.g., the shape of handwritten Arabic characters depends on their position in the word (at the beginning, in the middle, at the end or isolated characters) and, moreover, some characters are provided with group of dots. These characteristics were used by Amin et. al. to elaborate "an efficient and original method for the recognition of isolated words". (Currently, the same group of authors is involved in a new system using syntactic methods to recognize words within sentences). In their research they combined statistical and syntactic methods.

In general, a cursive word is recognized through a hierarchical analysis: a word is decomposed into letters, letters into strokes and strokes into primitive elements [Eden'81]. The hierarchical analysis can also be made with the aid of geometrical graph representation that models the line structures in a given cursive word [Hayes'79].

IRAC (Interactive Recognition of Arabic Characters), developed by Amin et. al., is a system capable of translating simple handwritten Arabic sentences into French. Two methods were experimented for the recognition of the words: The former is a syntactical method based on the segmentation of words into primitives like curves and strokes. An automaton transforms the range of primitives into the list of characters constituting the word. The latter method uses a global approach: each word is identified according to a vector of some pre-determined parameters. Then a syntactical and semantical analyzer verifies the grammatical structure and the meaning of the Arabic sentence which is translated word by word into French. The French sentence is pronounced by a speech synthesizer.

4.3 INDUSTRIAL APPLICATIONS OF PR

The industrial applications of pattern recognition are so enormous that one can not even attempt to enumerate them. In the following sections only some of these applications will be mentioned.

4.3.1 MACHINE VISION AND ROBOTICS

Although robot vision is a new technology, it has been applied successfully to a variety of industrial needs in inspection, manufacturing, and material handling [Trombly '84]. New developments in robot technology are creating practical, cost effective solutions for a variety of industrial needs.

A year or two ago, researchers and robot manufacturers interested in implementing a robot vision application could take one of two

approaches [Trombly and Hudson '84]. The first approach was to purchase all the necessary vision components from various sources. That meant buying an image processor from one company, a camera from another and lens and light sources from yet another. The user then had to assemble the pieces, and in most instances he had to write all of his own software to test, analyze and process the vision application. The second and most common approach was to contract with the vision equipment vendor for the development and installation of a turnkey inspection or manufacturing system. The robot user and his company paid a premium for their vision system in an effort to assure the success of the system.

Since 1981, emphasis on robotics has skyrocketed [Trombly-1984]. New groups have been formed in many manufacturing companies with the determination to learn about, test and initially apply new robot and automation technologies. Machine vision is one of the new technologies being tested and applied. This focused interest has created a need for a robot vision system that makes it easy for manufacturing engineers to learn about, test, and implement a robot vision application. Trombly and Hudson addressed these needs in a newly developed vision system:

Vision Development System (VDS) is a complete hardware and software product for the development and testing of robot vision applications. A complimentary, low cost Target Application System (TASK) runs the application program developed with the VDS. An actual robot vision application that demonstrates inspection and

pre-assembly for keyboard manufacturing is used to illustrate the VDS/TASK approach. A detailed description of each stage of development and implementation was elaborated, including:

- a. defining the requirements,
- b. matching the application to the vision system,
- c. developing the application software, and
- d. building and installing the Vision System.

4.3.2 DEFECT DETECTION IN TEXTILES

Ade et. al. 1994 described a general system for automatic inspection of textured surfaces. The system is applicable, for example, to the industrial automatic inspection of textile fabrics. A normal such web is characterized by a fairly regular and visually homogeneous texture. Any deviation of the structure from normal, exceeding a certain tolerance threshold, is to be detected on-line. The main difficulty in such an application, according to Ade, is to select a set of texture measures capable of adequate representation of the texture and of efficient computability. Measurement of textural neighborhood properties can be achieved using local matches. The resulting system can be looked upon as a bank of filters. A relevant thorough investigation was carried out by Laws previous to the work of Ade. He proposed the characterization of a texture by a set of "texture energy measures" computed by squaring and combining the outputs of a set of filters. Ade extended this approach by the proposal to use first order statistics associated with the coefficients of a sliding local linear transform for texture analysis and characterization. From the results of Ade approach,

the filter sets obtained compare, in general, favorably with the set of empirical filters introduced by Laws.

4.3.3. A PB SYSTEM FOR THE RE-IDENTIFICATION OF MOTOR VEHICLES

In order to obtain reliable data for traffic management and control, it is necessary to re-identify vehicles after passing a specific section of a road network. This allows the measurement of journey times, traffic density, etc.

For this purpose a pattern recognition system was developed by Pfannerstill [1984]. Vehicles are characterized by signals of an inductive loop detector. Information contained in these is compressed on a few features by means of the Karhunen-Loeve expansion. This compression is necessary in order to minimize the capacity required for data paths to transmit these features and computer power to correlate the objects with each other.

A simple metric is used to re-identify single vehicles. Correlation analysis is performed for the identification of platoons of vehicles in order to measure the above mentioned traffic data. The result is a pattern recognition system with only minimal requirements for (1) data paths to transmit the selected features and (2) computer capacity for the correlation.

BIBLIOGRAPHY

ORIGINAL PAGE IS
OF POOR QUALITY

- [11. Abdelmalek, N.M., Kasvand, T., Goupil, D., "Fingerprint Data Compression.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE, Vol.2, 1984, pp. 934-36.
- [21. Abe, K., Kondo, Y., Fukumura, T., "Character Recognition By Writing Rules On Strings Of Chain Codes.", Info. Process. Soc. JPN. (Joho Shori), Japan, Vol. 19, No.5, 1979, pp. 444-50.
- [31. Ade, Frank Lina, Norbert Unser, Michael, "Comparison Of Various Filter Sets For Defect Detection In Textiles.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE, Vol.1, 1984, pp. 429-30.
- [41. Ahmed, P., Suan, C.Y., "Edge Classification And Extraction Of Shape Features.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE, Vol. 1, 1984, pp. 593-6.
- [51. Ali, F., Pavlidis, T., "Syntactic Recognition Of Handwritten Numerals.", IEEE Trans. Syst. Man Cybern. SMC-7, No.7, 1977, pp. 537.
- [61. Amin, A., Masini, G., Haton, Jean-Paul, "Recognition Of Handwritten Arabic Words And Sentences.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE, Vol.2, 1984, pp. 1055-57.
- [71. Arduino, P., de Mori, R., Ghione, E.G., Piccolo, E., "Computer Interpretation Of Carotid Waves Using A Knowledge Based System.", Signal Processing, (Netherlands) Vol., 4, No. 5-6, 1982, pp. 409-23.
- [81. Bjorklund, C.M., Pavlidis, T., "On The Automatic Inspection And Description Of Printed Wiring Boards.", Proc. Of The International Conference On Cybernetics And Society, Washington, D.C., USA, New York, USA, Vol. XVII&779, 1977, 690-3., IEEE, 19-21 Sept.
- [91. Blacke, R.E., "An Approach To Multi-Sensor Syntactic Pattern Recognition.", Munich, Germany., New York, USA, 2 Vol. XXIX&1241, Vol.1, No.9, 1982, pp. 175-7, IEEE, 19-22 Oct.
- [101. Blacke, R.E., "Software Support For The Construction Of Syntactic Pattern Recognition By A Research Engineer." Pattern Recognition (GB), 1980 Conference On Pattern Recognition, Vol.14, No.1-6, 1981, pp. 205-10, Oxford, England, 9-11 Jan. 1980.

ORIGINAL PAGE IS
OF POOR QUALITY

- [111. Blacke, R.E., "A Weak Algorithm For The Inference Of String And Tree Grammars.", Proc. Of The 5th Int. Conf. On Pattern Recognition., Miami Beach, Fl. USA. New York, USA. 2 Vol. XLVI:1392., 1990, pp.999-1000., Int. Assoc. Pattern Recognition, IEEE. 1-4 Dec. 1980.
- [121. Bourne, J.R. et. al., "Evaluation Of A Syntactic Pattern Recognition Approach To Quantitative Electroencephalographic Analysis.", *Electroencephalogram And Clin. Neurophysiol.* (Netherlands). Vol.52. No.1. 1981 pp. 57-64.
- [131. Bozinovic, Radmilo et.al., "Knowledge-Based Cursive Script Interpretation.", Proc. 7th. Int. Conf. On Pattern Recognition., Montreal, Canada. IEEE, Vol.2, 1984, pp.774-776.
- [141. Praccini, C. Gambardella, G. Grattarola, A., "Shift-Variant Image Processing For Scale-Invariant Recognition", Proc. SPIE Int. Soc. Opt. Eng. (USA).. Application Of Digital Image Processing., Vol.397. Geneva, Switzerland. 1983, pp. 319-25., 11. SPIE. 19-22 April 1983.
- [151. Bronson, E.C., "Syntactic Pattern Recognition Of Discrete Utterances.", Proc. Of ICASSP 83. IEEE Int. Conf. On Acoustics, Speech And Signal Processing. Boston, MA., USA. New York, USA. 3 Vol. 1983, pp. 719-22 Vol.2. IEEE. 14-16 April 1983.
- [161. Bunke, H., "Experience With Several Methods For The Analysis Of Schematic Diagrams.", Proc. Of The 6th Int. Conf. On Pattern Recognition. Munich, Germany., New York, USA. 2 Vol. XXIX:1241. Vol.2. No.9, 1992, pp.710-12., IEEE. 19-22 Oct. 1992.
- [171. Bunke, H., "Attributed Programmed Graph Grammars And Their Application To Schematic Diagram Interpretation.", *IEEE Trans. Pattern Anal. And Mach. Intell.* (USA).. Vol. PAMI-4, No.4., 1992, pp. 574-92.
- [181. Burns, J.B. Hanson, A.R. Riseman, E.M., "Extracting Straight Lines.", Proc. 7th Int. Conf. On Pattern Recognition., Montreal, Canada. IEEE, Vol.1. 1984, pp.482-485.
- [191. Chang, N.S., Fu, K.S., Onoe, M., Preston, K. Jr., Rosenfeld, A., "A Study On Parallel Parsing Of Tree Languages And Its Application To Pattern Recognition.", Proc. Of Part Of The Japan-United States Seminar On Research Towards Real-time Parallel Image Analysis And Recognition. Tokyo, Japan. New York, USA. 1981.

ORIGINAL PAGE IS
OF POOR QUALITY

- [201. Cheng, N.S., Fu, K.S., "Parallel Parsing Of Tree Languages For Pattern Recognition.", *Pattern Recognition (GB)*, Vol.11, No.3, 1979, pp.213-22.
- [211. Chatterji, B.N., Sarkar, D., "Toward A Feature Extraction Scheme For ECG Recognition.", *Proc. Of 6th Int. Conf. On Pattern Recognition*, Munich, Germany., New York, USA, 2 Vol. XXIX&1241., Vol.2, No.5, 1992, pp.988-90. IEEE, 19-22 Oct. 1992.
- [221. Chen, C.H., "Pattern Recognition And Signal Analysis", *Pattern Recognition And Signal Processing*, Paris, France, Alphen Aan Den Rijn, Netherlands, Sijthoff And Noordhoff., 25 June - 4 July 1978.
- [231. Chiang, Y.T., Fu, K.S., "Parallel Parsing Algorithms And VLSI Implementations For Syntactic pattern Recognition", *IEEE Trans. Pattern Anal. And Mach. Intell.* USA., Vol. PAMI-6, No.3, 1984, pp. 302-14.
- [241. Chiang, Y.T., Fu, K.S., "VLSI Architectures For Syntactic Pattern Recognition.", *Proc. Of The IEEE Computer Society Conference On Computer Vision And Pattern Recognition*, Washington, D.C., USA., Silver Spring, MD., USA. VII&559., 1993, pp. 474-81., 21., IEEE Comput. Soc. Proc. IEEE, 19-23 June 1993.
- [251. Chiang, Y., Fu, K.S., "Parallel Processing For Distance Computation In Syntactic Pattern Recognition." 1981 IEEE Comput. Soc. Workshop On Computer Architecture For Pattern Analysis And Image Database Management. Hot Springs, Va., USA., New York, USA. IX&349, 1981, pp. 337-344, 19. IEEE, 11-13 Nov. 1981.
- [261. Datta, Lokesh and Morgera, Salvatore, "Optimal Feature Selection: Part II - Implementation.", *Proc. 7th Int. Conf. On Pattern Recognition*, Montreal, Canada. IEEE Vol. 1, 1994, pp. 139-140.
- [271. Davies, E.R., "Estimation Of Edge * Orientation By Template Matching.", *Proc. 7th Int. Conf. On Pattern Recognition*, Montreal, Canada, IEEE Vol. 1, 1994, pp. 49-51.
- [281. Davis, L.S., Henderson, T.C., "Hierarchical Constraint Processes For Shape Analysis.", *IEEE Trans. Pattern Anal. And Mach. Intell. (USA)*, Vol. PAMI-3, No.3, 1981, pp. 265-77.
- [291. De, T.K. and Chatterji, B.N., "An Approach To A Generalised Technique For Image Enhancement.", *Proc. 7th Int. Conf. On Pattern Recognition*, Montreal, Canada, IEEE Vol. 1, 1994, pp. 604-06.

- [301. Dengler, Joashim and Bille, Josef. "A Scene Analysis System Simulating The Peripheral Visual Perception.". Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada. IEEE Vol.2, 1984, pp. 643-65.
- [311. Diday, E. Govaert, G. Lechevallier, Y. Sidi, J., "Clustering In Pattern Recognition.". Proc. Of The 5th Int. Conf. On Pattern Recognition, Miami Beach, Fl. USA., New York, USA. 2 Vol. XLVI&1392., 1980. pp. 424-P., IEEE Int. Assoc. Pattern Recognition, 1-4 Dec. 1980.
- [321. Doster, Wolfgang, "Different States Of A Document's Content On Its Way From The Gutenbergian World To The Electronic World.". Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol.2, 1984, pp. 872-74.
- [331. Duncan, James S., "Intelligent Detection Of Left Ventricular Boundaries In Gated Nuclear Medicine Image Sequences.". Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol.2, 1984, pp. 975-77.
- [341. Dutta Majumder, D. and Pal, S.K., "On Fuzzification, Fuzzy Language And Multi-Category Fuzzy Classifier.", Proc. On Int. Conference On Cybernetics And Society, Washington, D.C. USA., New York, USA. XVII&779., 1977, pp. 591-5. IEEE 19-21 Sept. 1977.
- [351. Dziernanowski, J. Shiavi, R. Bourne, J., Gerhard, G.C. Miller, W., "A Syntactic Method For Evaluation Of Abnormalities In Human Gait.". Frontiers Of Engineering And Computing In Health Care. Proc. Of The 5th Annual Conference, Columbus, OH, USA., New York, USA., 1983, pp. 12-14, IEEE. 10-12 Sept. 1983.
- [361. Enomoto, H. Yonfzaki, N. Satoh, K., "Statistical Approach For Construction And Estimation Of Syntactic Pattern Recognition Mechanism.". Proc. Of The 4th Int. Joint Conference On Pattern Recognition., Kyoto, Japan. XXII&1166., 1979. pp.359-63., Kyoto Univ. Internat. Assoc. Pattern Recognition. 7-10 Nov. 1979.
- [371. Ferber, G., "Syntactic Pattern Recognition In Clinical Routine EEG.". Proc. Of The 5th Int. Conference On Pattern Recognition, Munich, Germany., New York, USA., Vol. XXIX&1241, 1196 Vol. 2 IEEE 19-22 Oct. 1982.
- [381. Filippaki, A.J., "A Least Mean-Squared Error Approach To Syntactic Classification.", IEEE Trans. Pattern Anal. And Mach. Intell. (USA). Vol. PAMI-2 No.3, 1980. pp. 252-255.

ORIGINAL PAGE IS
OF POOR QUALITY

- [391. Fliess, M., "A Functional Theory Of The Realization Of Sampled Recursive Multidimensional Filtering.", *Inf. And Control (USA)*, Vol 43, No. 3, 1978, pp. 339-55.
- [401. Fu, K.S., "VLSI Architectures For Syntactic Pattern Recognition Algorithms.", *IEEE Workshop On Languages For Automation*, Chicago, IL, USA., Silver Spring, MD, USA., VI&311 217-19 IEEE Comput. Soc. Press., IEEE 7-9 Nov. 1983.
- [411. Fu, K.S., Bekey, G.A., Saridis, G.N., "Attributed Grammars For Image Analysis.", *Identification And System Parameter Estimation*, Proc. Of The 4th IFAC Symposium, Washington, D.C., USA., Oxford, England, Vol. XVII&1709, 1983, pp.1449-54, Vol.2, 37 Pergamon, IFAC 7-11 June 1982.
- [421. Fu, K.S., "A Step Towards Unification Of Syntactic And Statistical Pattern Recognition.", *IEEE Trans. Pattern Anal. And Mach. Intell. (USA)*, Vol. PAMI-5, No.2, 1983, pp.200-5.
- [431. Fu, K.S., "Pattern Recognition For Automatic Visual Inspection.", *Proc. SPIE Int. Soc. Opt. Eng. (USA)*, Robot Vision, Vol. III, Arlington, Va. USA., 1982, pp. 12-19, SPIE, 6-7 May 1982.
- [441. Fu, K.S., "Attributed Grammars For Pattern Recognition - A General (syntactic-Semantic) Approach.", *Proceedings Of PRIP 82*, IEEE Comput. Soc. Conference On Pattern Recognition And Image Processing., Las Vegas, NV, USA., New York, USA. XII&701., 1982, pp. 19-27, IEEE 14-17 June, 1982.
- [451. Fu, K.S., Kanal, L.N., Rosenfeld, A., "Recent Progress In Syntactic Pattern Recognition.", *Progress In Pattern Recognition*, Vol.1 Amsterdam, Netherlands., 1981, pp. 1-31, 62, North-Holland.
- [461. Fu, K.S., Gelsema, E.S., Kanal, L.N., "Syntactic Models In Pattern Recognition And Applications.", *Pattern Recognition In Practice*, Proc. Of An Int. Workshop, Amsterdam, Netherlands., Amsterdam, Netherlands. XII&552, 1980, pp. 303-14., North-Holland, 21-23 May 1980.
- [471. Fu, K.S., "Recent Developments In Pattern Recognition." *IEEE Trans. Comput. (USA)*, Vol. C-29, No.10, 1980, pp. 845-56.
- [481. Fu, K.S., "Syntactic Pattern Recognition And Application", *Computer Conference Spring 1980*, VLSI: New Architectural Horizons, San Francisco, CA, USA., New York, USA. XVI&504, 1980, pp. 391-7., IEEE 25-28 Feb. 1980.

ORIGINAL PAGE IS
OF POOR QUALITY

- [491. Fu, K.S., "Recent Advances In Syntactic Pattern Recognition Proc. Of The 4th Int. Joint Conf. On Pattern Recognition, Kyoto, Japan, Kyoto, Japan, XXIII:1144, 1978, pp. 13-19 Kyoto Univ. Internat. Assoc. Pattern Recognition, 7-10 Nov. 1978.
- [501. Fu, K.S., Chen, D.H., "Syntactic Pattern Recognition And Its Application To Signal Processing.", Pattern Recognition And Signal Processing., Paris, France. Alphen Aan Den Rijn, Netherlands. XI:455., 1979, pp. 1-21 Sijthoff And Noordhoff 25 June - 4 July 1979.
- [511. Fu, K.S. and Winston, A., "Pattern Recognition: Theory And Application.", Groningen, Netherlands. VIII:501. Noordhoff., 1977. pp. 99-132.
- [521. Fu, K.S. and Rosenfeld, A., "Pattern Recognition And Image Processing.", IEEE Trans. Comput. (USA), Vol. C-25, No. 12, 1976, pp. 1336-46.
- [531. Fu, K.S. and Booth, T.L., "Grammatical Inference - Introduction And Survey.", IEEE Trans. Syst. Man Cybern., Part I and Part II, SMC-5, 1975, 409.
- [541. Funakubo, Noboru, "Region Segmentation Of Biomedical Tissue Image Using Color Texture Feature.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada. IEEE, Vol. 1, 1984, pp. 30-32.
- [551. Furst, M.A. and Gaines, P.E., "Edge Detection For Digital Grey Level Image Via Dynamic Programming.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada. IEEE Vol. 1, 1984, pp. 55-58.
- [561. Gabry, G.E. and Anderson, K.R., "Primitive Extraction For A Syntactic Pattern Recognizer Of Features In Seismic Signals.", Proc. Of ICASSP 83. IEEE Int. Conf. On Acoustic, Speech And Signal Processing., Boston, MA, USA. New York, USA., Vol.3, No. 4., 1983, pp.993-6 IEEE 14-16 April 1983.
- [571. Gelsema, E.S. and Kanal, L.N., "Pattern Recognition In Practice.", Proc Of An Int. Workshop. Amsterdam, Netherlands. XII:552. North-Holland. 21-23 May 1980.
- [581. Gelsema, E.S., "Pattern Recognition.", Informatieve (Netherlands). Vol.25, No.6, 1983, pp.58-64.
- [591. Geyns, A.S., "Pattern Recognition Of Human Brain Electrical Potentials.", IEEE Trans. Pattern Anal. And Mach. Intell. (USA) Vol. PAMI-2, No.5, pp. 393-404.

**ORIGINAL PAGE IS
OF POOR QUALITY**

- [601. Giese, D.A., Bourne, J.R., Ward, J.W., "Syntactic Analysis Of The electroencephalogram.", IEEE Trans. Syst. Man Cybern. (USA). Vol. SMC-9, No.9, 1979, pp.429-35.
- [611. Gilmore, J.F. and Pemberton, W.B., "A Survey Of Aircraft Classification Algorithms.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada IEEE Vol. 1, 1984, pp. 559-561.
- [621. Gilmore, J.F., "Image Registration Through The Exploitation Of Perspective Invariant Graphs.", Proc. SPIE Int. Soc. Opt. Eng. (USA). Applications Of Digital Image Processing., Vol. 397 Geneva, Switzerland, 1983, pp. 55-63., SPIE, 19-22 April 1983.
- [631. Gilmore, J.F., "Syntactic Pattern Recognition Approach To Scene Matching.", Proc. SPIE Int. Soc. Opt. Eng. (USA), Applications Of Digital Image Processing, IV, Vol. 359 San Diego, CA, USA, 1983, pp. 259-65, 24-27 Aug. 1982.
- [641. Gindy, Gene R. and Gmitro, Arthur F., "Optical Feature Extraction Via The Radon Transform.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE, Vol.2, 1984, pp. 702-04.
- [651. Goldfarb, Lev and Chan, Tony Y.T., "On A New Unified Approach To Pattern Recognition.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol.2, 1984, pp. 705-08.
- [661. Granger, C., " Symbolic Scene Matching.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol.2 1984, pp. 883-85.
- [671. Graemuller, Hans et. al., "Local Feature Extraction For Model-Based Workpiece Recognition.", Proc. 7th Int. Conf. On Pattern Recognition., Montreal, Canada, IEEE Vol.2, 1984, pp. 886-89.
- [681. Gyimothy, T. and Dombi, J., "Syntactic Pattern Recognition With Modified Fuzzy Automata.", Comput. Linguist. And Comput. Lang. (Hungary). No. 15, 1982, pp. 175-193.
- [691. Hanakata, K., "Feature Selection And Extraction For Decision-Theoretic Approach And Structural Approach.", Noordhoff International, Leyden, The Netherlands, 1977.
- [701. Haralick, R.M., "Structural Pattern Recognition. ", Homomorphism And Arrangements., Pattern Recognition (GB). Proc. Of The IEEE Comput. Soc. Conference On Pattern Recognition And Image Processing, 1977, Vol. 10, No. 3, Troy, NY, USA, pp. 223-36, 6-9 June, 1977.

- [711. Harlow, C.A., Connors, R.W., Trivedi, M., "A Computer Vision System For The Analysis Of Aerial Scenes.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1, 1984, pp. 407-410.
- [721. Hatten, M.L., Fu, K.S., Whinston, A., "A Systems View Of Strategy Formulation In Social Choice Via Syntactic Pattern Recognition.", Pattern Recognition: Theory And Application., Groningen, Netherlands., 1977, pp. 397-415. Noordhoff.
- [731. Hausser, Rainer F., "A Stochastic Approach To Edge Detection., Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1, 1984, pp. 52-54.
- [741. Henderson, T.C. and Davis, L., "Hierarchical Models And Analysis Of Shape.", Pattern Recognition (69).. 1980 Conference On Pattern Recognition, Vol.14, No. 1-4 Oxford, England, pp. 197-204 9-11 Jan. 1980
- [751. Hernan, M.A., Jimenez, J., Lavington, S., "Image Analysis Of Experimental Data In Turbulance.", Info. Processing 80. Proc. Of The IFIP Congress 80. Tokyo, Japan. Amsterdam, Netherlands. XIII:1970. pp. 753-6.. North-Holland. 6-9 Oct. 1980.
- [761. Hernandez, C. and Arias, J.E., "Syntactic Pattern Recognition Of Fetal Stress.", Journal Of Biomedical Engineering, Vol.1, No.6, April 1984.
- [771. Ichino, M. and Sklansky, J., "Feature Selection For Linear Classifiers.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1, 1984, pp. 124-127.
- [781. Ikeuchi, Katsushi, " A Model Character Recognition By Humans.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1, 1984, pp. 521-524.
- [791. Ioannidis, A., Kazakos, D., Watson, D., "Application Of Median Filtering On Nuclear Medicine Scintigram Images." Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1, 1984, pp. 33-36.
- [801. Jansen, B.H., Bourne, J.R., Ward, J.W., "Feature Extraction Methods For Syntactic Pattern EEG Analysis." IEEE 1980 Frontiers Of Engineering In Health Care, Washington, DC, USA., New York, USA, IX:371., pp. 231-234. IEEE 29-30 Sept. 1980.

ORIGINAL PAGE IS
OF POOR QUALITY,

- [911. Kanal, L.N., "Parsing Of Medical Waveforms (Pattern Recognition)". EASCON '80 Record. IEEE Electronics And Aerospace Systems Conventions. Arlington, Va, USA. New York, USA., 1980, p. 554. IEEE 29 Sept. - 1 Oct. 1980.
- [921. Kanal, L.N., "Problem-Solving Models And Search Strategies For Pattern Recognition.", IEEE Trans. Pattern Anal. And Mach. Intell. (USA)., 1978 IEEE Computer Soc. Workshop On Pattern Recognition And Artificial Intelligence, Vol.PAMI-1, No.2. Princeton, NJ, USA., pp. 193-201, IEEE, 12-14 April 1978.
- [931. Keng, J., Fu, K.S., "A System Of Computerized Automatic Pattern Recognition For Remote Sensing.", Proc. 1977 Int. Computer Symposium. Taipei, Taiwan, 27-29 Dec. 1977.
- [941. Keng, J. and Fu, K.S., "A Syntax-Directed Method For Land-Use Classification Of LANDSAT Images.", Proc. Symp. Curr. Math. Probl. Image Sci., Monterey, Calif. Nov. 10-12, 1976, pp. 261-65.
- [951. Khanmamedov, Oktai K., "Algebraic Theory Of Pattern Recognition", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada. IEEE Vol. 2, 1984, pp. 914-16.
- [961. Kayser, K. and Hoffgen, H., "Pattern Recognition In Histopathology By Orders Of Textures.", Med. Info. (GB). Vol.9, No. 1, 1984, pp. 55-9.
- [971. Kot, L., "A Syntax-Controlled Segmentation Of Speech Signal On The Basis Of Dynamic Spectra.", Proc. Of ICASSP 82. IEEE Int. Conf. On Acoustics, Speech And Signal Processing. Paris, France. New York, USA., pp. 2015-2017. IEEE, 3-5 May 1982.
- [981. Kubota, K., Iwaki, O., Arakawa, H., "Document Understanding System.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada. IEEE Vol. 1, 1984, pp. 612-614.
- [991. Kyun, Kim Tae and Takeshi, Agui, "A Study On The Pattern Recognition Of Korean Characters By Syntactic Methods.", J. Korean Inst. Electron. Eng. Vol. 14, No. 5, 1977, pp. 15-21.
- [901. Lamarche, Francois and Plamondon, Rejean, "Segmentation And Feature Extraction Of Handwritten Signature Patterns." Proc. 7th Int. Conf. On Pattern Recognition. Montreal, Canada. IEEE Vol. 2, 1984, pp. 756-59.
- [911. Lang, M., "Pattern Recognition.", Ephys. Bl. (Germany). Vol. 39, No. 11, 1993, pp. 367-72.

ORIGINAL PAGE IS
OF POOR QUALITY

- [92]. Le Chevalier, F., Bobilot, G., Fugier-Garrel, C., "Radar Target And Aspect Angle Identification.", Proc. Of The 4th Int. Joint Conference On Pattern Recognition. Kyoto, Japan. XXII&1166., 1978; pp. 398-400. Kyoto, Univ. Internat. Assoc. Pattern Recognition. 7-10 Nov. 1978.
- [93]. Leclerc, Yvan and Zucker, Steven, "The Local Structure Of Image Discontinuities In One Dimension.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1, 1984, pp. 46-48.
- [94]. Lee, E.T., "Fuzzy Tree Automata And Syntactic Pattern Recognition.", IEEE Trans. Pattern Anal. And Mach. Intell. (USA). Vol. PAMI-4, No. 4, 1982, pp. 445-9.
- [95]. Levine, B., "The Inference Of Tree Systems Using Tree Derivatives And A Sample Strength Parameter.", Proc. Of The 5th Int. Conf. On Pattern Recognition. Miami Beach, FL, USA. New York, USA. Vol. XLVI&1382., 1980, pp. 991-4. IEEE Int. Assoc. Pattern Recognition. 1-4 Dec. 1980.
- [96]. Li, Zhong-rong, "A Fingerprint Recognition System With Micro-Computer.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 2, 1984, pp. 939-941.
- [97]. Liu, H.H. and Fu, K.S., "A Syntactic Pattern Recognition Approach To Seismic Discrimination.", Geop Exploration (Netherlands). Vol. 20, No. 1, 1982, pp. 183-96.
- [98]. Liu, H.H. and Fu, K.S., "An Application Of Syntactic Pattern Recognition To Seismic Discrimination.", IEEE Trans. Geosci. And Remote Sensing USA. Vol GE-21, No.2 1983, pp. 125-32.
- [99]. Liu, Hsi-Ho and Fu, K.S., "A Syntactic Approach To Seismic Pattern Recognition.", IEEE Trans. Pattern Anal. And Mach. Intell. (USA). Vol. PAMI-4, No. 2, 1982, pp.136-140.
- [100]. Liu, H.H. and Fu, K.S., "An Application Of Syntactic Pattern Recognition To Seismic Discrimination.", Proc. On 2nd Int. Symposium On Computer Aided Seismic Analysis And Discrimination., North Dartmouth, MA, USA. New York, USA. VI&155. 1981, pp. 49-53. IEEE 19-21 Aug.1981
- [101]. Loy, W.W. Landau, I.D., "An On-Line Procedure For Recognition Of Handwritten Alpha-Numeric Characters.", Proc. Of The 5th Int. Conf. On Pattern Recognition. Miami Beach, FL, USA. New York, USA. Vol. XLVI&1382., 1980, pp. 712-14, IEEE Int. Assoc. Pattern Recognition. 1-4 Dec. 1980.

- [102]. Lu, Shin-Yee Fu, K.S., "Error-Correcting Tree Automata For Syntactic Pattern Recognition.", IEEE Trans. Comput. (USA). Vol. C-27, 1978, No. 11 pp. 1040-53.
- [103]. Lu, S.Y. Chen, C.H., "Error-Correcting Matching And Parsing For Syntactic Pattern Recognition.", Pattern Recognition And Signal Processing. Paris, France. Alphin Aan Den Rijn, Netherlands. XI&655. pp. 23-39. Sijthoff And Noordhoff 25 June - 4 July 1978.
- [104]. Lu, S.Y. and Fu, K.S., "A Sentence-To-Sentence Clustering Procedure For Pattern Analysis.", IEEE Trans Syst. Man Cybern., SMC-8 (5), p.381, 1978.
- [105]. Lu, S.Y. and Fu, K.S., "Stochastic Error-Correcting Syntax Analysis For Recognition Of Noisy Patterns.", IEEE Trans. Comput., C-26,(12), p.1268, 1977.
- [106]. Lu, S.Y. Fu, K.S., "Structure-Preserved Error-Correcting Tree Automata For Syntactic Pattern Recognition." Proc. Of The 1976 IEEE Conference On Decision And Control Including The 15th Symposium On Adaptive Processes. Clearwater, FL, USA. New York, USA. 1976, XXIV&1313. pp. 413-19. IEEE 1-3 Dec. 1976.
- [107]. Miclet, L., "Inference For Regular Expressions.", 3rd Int. Joint Conference On Pattern Recognition., Coronado, CA, USA. New York, USA. XII&884, pp. 100-5. IEEE ACM IFIP et. al. 9-11 Nov. 1976.
- [108]. Moore, R.K., "A Multilevel Approach To Pattern Processing" Pattern Recognition (GB). 1980 Conference On Pattern Recognition. Vol. 14, No. 1-6 Oxford, England. pp. 261-5 9-11 Jan. 1980.
- [109]. Morgera, Salvatore Datta, Lokesh, "Optimal Feature Selection: Part I - Theory.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1, 1984. pp. 134-137.
- [110]. Mrchev, Simeon-Jord, "Reproducing And Recognition At Human Instantaneous Memory.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 2, 1984. pp. 968-71.
- [111]. Muehlenfeld, E. and Raubenheimer, H.R., "Image Processing With Matched Scanning Of Contours.", Proc. SPIE Int. Soc. Opt. Eng. (USA). Application Of Digital Image Processing. Vol. 397 Geneva, Switzerland. pp. 125-30 SPIE 19-22 April 1983.

- [112]. Mundy, J.L., "The Application Of Syntactic Pattern Recognition To Industrial Automation.", 1978 Joint Automatic Control Conference. Philadelphia, PA, USA. Pittsburg, PA, USA. pp. 315-19 ISA IEEE et al. 15-20 Oct. 1978.
- [113]. Neugebauer, G., "The Textscanner: A Model For Analyzing And Classifying Document Elements.", Proc. Of The 6th Int. Conf. On Pattern Recognition. Munich, Germany. New York, USA. Vol. XXIX&1241. pp. 486-8. IEEE 19-22 Oct. 1982.
- [114]. Nguyen, P.T. Asfar, L. Simon, S., "An Application Of Pattern Recognition For Structural Analysis In Geology." Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 2, 1984, pp. 875-77.
- [115]. Orouke, J. Supowit, K.J., "Some NP-HARD Polygon Decomposition Problems.", IEEE Trans. Inf. Theory (USA). Vol. IT-29, No. 2, 1983, pp. 181-90.
- [116]. Otsu, N., "Karhunen-Loeve Line Fitting And A Linearity Measure.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1, 1984, pp. 486-489
- [117]. Page, C. Filipski, A., "Discriminant Grammers, An Alternative To Parsing For Pattern Classification.", Proc. 1977 IEEE Workshop On Picture Data Description And Management.", IEEE Piscataway, N.J., 1977.
- [118]. Parui, S.K. Chaudhuri, B.B. Dutta Majumder, D., "A Procedure For Recognition Of Connected Handwritten Numerals.", Int. J. Syst. Sci. (GB). Vol. 13, No. 9, pp. 1019-29.
- [119]. Pavel, M., "Pattern Recognition: The Categorical Letting.", Inf. Sciences (USA). Vol. 17, No. 1, 1979, pp. 43-74.
- [120]. Pavlidis, T., "Syntactic Pattern Recognition On The Basis Of Functional Approximation.", 1978 Joint Workshop On Pattern Recognition And Artificial Intelligence. Hyannis, Mass, USA. New York, USA. VI&129. IEEE 1-3 June 1976.
- [121]. Pfannerstill, Elmar, "A Pattern Recognition System For The Re-identification Of Motor Vehicles.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1 1984, pp. 553-555.
- [122]. Pik, J., "A Hierarchical Pattern Description In The Syntactic Approach To Pattern Recognition.", Pattern Recognition (GB) 1980 Conference On Pattern Recognition. Vol. 14, No.1-6, Oxford, England. pp. 191-6 9-11 Jan. 1980.

- [123]. Prewitt, J.M.S., "Canonical Representation For Textures And Textures.", Proc. Of COMPSAC The IEEE Computer Society's 3rd Int. Computer Software And Applications Conference. Chicago, IL, USA. New York, USA. pp. 470-6 IEEE 6-8 Nov. 1979.
- [124]. Queiros, C.E. and Gelsema, E.S., "On Feature Selection." Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada IEEE Vol. 1, 1984, pp. 128-130.
- [125]. Richetin, M. and Vernadat, F., "Efficient Regular Grammatical Inference For Pattern Recognition.", Pattern Recognition. (GB). Vol. 17, No. 2, 1984, pp. 245-50.
- [126]. Rosenfeld, A. and Weszka, J.S., "Picture Recognition And Scene Analysis.", Computer, May 1976, pp.28-38.
- [127]. Samy, R.A. and Bozzo, C.A., "Dynamic Scene Analysis And Video Target Tracking.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada IEEE Vol. 2, 1984, pp.993-995.
- [128]. Schaefer, B.A., "A Model Of Human Symmetry Detection.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1, 1984, pp. 509-511.
- [129]. Scherl, W. and Hundt, E., "A Syntactic Method Of Pattern Recognition For Document Processing.", Siemens Forsch. - And Entwicklungsber., 1984, (Germany). Vol. 13, No.3 pp. 109-13.
- [130]. Sethi, I.K., "A General Scheme For Discontinuity Detection., Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1, 1984, pp. 42-44.
- [131]. Shi, Q.Y. and Fu, K.S., "Parsing And Translation Of (Attributed) Expansive Graph Languages For Scene Analysis. Proc. Of 6th Int. Conf. On Pattern Recognition. Munich, Germany. New York, USA. Vol. XXIX&1241., 1984 pp. 684-687. Vol. 2, IEEE 19-22 Oct. 1982.
- [132]. Shjouxuan, Z., "A Chinese Character Recognition System Based On Pictorial Database Technique.", Proc. Of The 6-th Int. Conf. On Pattern Recognition. Munich, Germany. New York, USA. Vol. XXIX&1241. pp. 780-2. Vol. 2. IEEE, 19-22 Oct. 1982.
- [133]. Siromoney, G. Siromoney, R. Subramanian, K.G., "Stochastic Table Arrays.", Computer Graphics And Image Processing (USA), Vol. 18, No. 2, 1982, pp. 202-11.
- [134]. Shapiro, L.G., "Inexact Matching Of Line Drawings In A Syntactic Pattern Recognition System.", Pattern Recognition (GB). Vol.10, No.5-6, 1978, pp. 313-21.

- [1351]. Stallings, W., "Fuzzy Set Theory Versus Bayesian Statistics.", IEEE Trans. Syst. Man And Cybern. (USA). Vol. SMC-7, No. 3, 1977, pp. 216-19.
- [1361]. Stockman, G.C. and Kanal, L.N., "Problem Reduction Representation For The Linguistic Analysis Of Waveforms.", IEEE Trans. Pattern Anal. And Mach. Intell. (USA). Vol. PAMI-5, No.3., 1983, pp. 287-98.
- [1371]. Strawn, J., "Approximation And Syntactic Analysis Of Amplitude And Frequency Functions For Digital Sound Synthesis.", Comput. Music J. (USA). Vol.4, No.3, 1980 pp. 3-24.
- [1381]. Tai, J.W., "A Syntactic-Semantic Approach For Chinese Character Recognition.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada IEEE Vol. 1, 1984, pp.374-376.
- [1391]. Tai, J.W. Fu, K.S., "Inference On A Class Of CFG By Means Of Semantic Rules.", Int. Joint Comput. And Inf. Sci. (USA). Vol. 11, No. 1, 1982, pp. 1-23.
- [1401]. Tai, J.W., "Inference Of A Class Of Context-Free Programmed Grammars For Syntactic Pattern Recognition.", IEEE Computer Society Conference On Pattern Recognition And Image Processing. Dallas, TX, USA. New York, USA. X&25. pp. 18-20. IEEE 3-5 Aug. 1981.
- [1411]. Tai, J.W., "Attributed Parallel Tree Grammars And Automata For Syntactic Pattern Recognition.", Proc. Of The 5th International On Pattern Recognition. Miami, FL, USA. New York, USA. Vol. XLVI&1382. 1001-3. IEEE Int. Assoc. Pattern Recognition. 1-4 Dec. 1980.
- [1421]. Tang, G.Y. and Huang, T.S., "A Syntactic-Semantic Approach To Image Understanding And Creation.", IEEE Trans. Pattern Anal. And Mach. Intell. Vol. PAMI-1, No.2 Princeton, NJ,USA., pp. 135-44. IEEE 12-14 April 1978.
- [1431]. Taniguchi, R. Yokota, M. Kawaguchi, E. Tamati, T., "Picture Understanding And Retrieving System Of Weather Chart.", Proc. Of The 6th Int. Conf. On Pattern Recognition. Munich, Germany. New York, USA. Vol. XXIX&1241. pp. 802-5. IEEE 19-22 Oct. 1982.
- [1441]. Tecuci, G.D., "H-Graphs And Their Applications To Pattern Recognition", Bul. Inst. Ploiteh. 'Gheorghe Gheorghiu-Dej' Bucuresti Ser. Electroteh. (rumania). Vol.43, No.3, 1981 pp. 23-34.
- [1451]. Thomason, M.G., "Syntactic/Semantic Pattern Recognition.", IEEE 1982 Proc. Of The Int. Conf. On Cybern. And Society. Seattle, WA., USA., New York, USA. pp. 520-3. IEEE 28-30 Oct.1982.

- [146]. Thomason, M.G., "Syntactic/Semantic Techniques In Pattern Recognition.", Int. J. Comput. And Info. Sci. (USA). Vol. 11, No. 2, 1982, pp. 75-100.
- M.G., "Stochastic Syntax-Directed Translations In Syntactic Pattern Processing.", Pattern Recognition (GB). 1980 Conference On Pattern Recognition. Vol. 14 No. 1-6 Oxford, England. pp. 187-90. 9-11 Jan. 1980.
- [147]. Thorne, R.E. Shwedyk, E.A. Seshia, S.S., Patrich, J. Cohen, B., "Syntactic Analysis Of Epileptic EEG.", IEEE 1981 Frontiers In Engineering In Health Care Conference. Houston, TX, USA., New York, USA. pp. 294-6 19-21 Sept. 1981.
- [148]. Trombly, John E. Hudson, David L., "A Practical Solution Using A New Approach To Robot Vision.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol.2 1984, pp. 1016-18.
- [149]. Tsai, W.H. Fu, K.S., "Subgraph Error-Correcting Isomorphism For Syntactic Pattern Recognition.", IEEE Trans. Syst. Man And Cybern. (USA)., Vol. SMC-13, No.1 1983, pp. 48-62.
- [150]. Tsai, W.H. and Fu, K.S., "A Syntactic-Statistical Approach To Recognition Of Industrial Objects.", Proc. Of The 5th Int. Conf. On Pattern Recognition. Miami, FL, USA. New York, USA. Vol. XLVI:1382. pp. 251-9. IEEE Int. Assoc. Pattern Recognition. 1-4 Dec. 1980.
- [151]. Tsai, W.H. and Fu, K.S., "Attributed Grammar - A Tool For Combining Syntactic And Statistical Approaches To Pattern Recognition.", IEEE Trans. Syst. Man And Cybern. (USA). Vol. SMC-10. No.12, 1980, pp. 873-85.
- [152]. Tsai, W.H. Fu, K.S., "A Pattern Deformational Model And Bayes Error-Correcting Recognition System.", IEEE Trans. Syst Man And Cybern. (USA). Vol. SMC-9 No.12, 1979, pp. 745- 56.
- [153]. Udupa, J.K. Murthi, I.S.N., "Syntactic Approach To EGG Rythm Analysis.", IEEE Trans. Biomed. Eng. (USA). Vol. BME-27, No.7, 1980, pp. 370-5.
- [154]. Veillon, F., "Towards A Systematic Study Of Shape Measures.", Proc. The 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1, 1984, pp. 607-610.

- [1553]. Vernadat, F. Richetin, M. Gros, J.B. Bekey, G.A. Saridis, G.N., "Characterization Of Dynamical Systems Via Syntactic Pattern Recognition.", Identification And System Parameter Estimation., 1982 Proc. Of The 6th IFAC Symposium. Washington, DC, USA. Oxford, England. Vol. XVII&1709. pp. 343-8 Vol.1 Pergamon. IFAC 7-11 June 1982.
- [156]. Vernadat, F. Richetin, M. Rives, G., "Regular Grammatical Inference By A Successor Method.", Proc. Of The 6th Int. Conf. On Pattern Recognition. Munich, Germany. New York, USA. Vol. XXIX&1241. pp. 688-91. Vol.2 IEEE 19-22 Oct. 1982.
- [157]. Verschueren, W. et. al., "Structural Recognition Of Handwritten Numerals.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada IEEE Vol. 2, 1984, pp. 760-762.
- [158]. Wong, A.K.S. and Chiu, D.K.Y., "A Probabilistic Inference System.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1, 1984 pp. 303-306.
- [159]. Xuan, Guorong, "A New Feature Selection Method Based On The Mahalanobis Distance.", Proc. 7th Int. Conf. On Pattern Recognition, Montreal, Canada, IEEE Vol. 1, 1984 pp. 131-133.
- [160]. You, K.C. and Fu, K.S., "A Syntactic Approach To Shape Recognition Using Attributed Grammars.", IEEE Trans. System, Man, And Cybernetics, Vol. SMC-9, No. 6, 1979, pp. 334-345.

BOOK REFERENCES

- [11. Aho, A.V., Ulman, J.D., The Theory Of Parsing, Translation And Compilation. Vol. 1, Prentice-Hall, Englewood Cliffs, N. J., USA, 1972.
- [21. Batchelor, B.G. Pattern Recognition: Ideas In Practice, Plenum Press, New York, USA. London, UK, 1978.
- [31. Bezdek, J.C., Pattern Recognition With Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
- [41. Caianiello, E.R., Musso, G., Cybernetic Systems: Recognition, Learning, Self-Organization, Research Studies Press, Ltitchworth, Hertfordshire, England, 1984.
- [51. Chen, C.H., Pattern Recognition And Artificial Intelligence, Academic Press, London, England, 1976.
- [61. Chen, C.H., Statistical Pattern Recognition, Hayden Book Co., Rochelle Park, N.J., 1973.
- [71. Chien, Y.T., Interactive Pattern Recognition, Marcel Dekker, New York, 1978.
- [81. Duda, R.O., Hart, P.E., Pattern Classification And Scene Analysis, John Wiley & Sons, New York.
- [91. Fu, K.S., Pattern Recognition And Applications, Englewood Cliffs, N.J., USA, 1982.
- [101. Fu, K.S., Applications Of Pattern Recognition, CRC Press, Boca Raton, Fl., USA, 1982.
- [111. Fu, K.S., Syntactic Methods In Pattern Recognition, Academic Press, New York, USA, 1974.
- [121. Fu, K.S., Syntactic Pattern Recognition Applications, Springer-Verlag, Basel, 1977.
- [131. Fukunaga, K., Introduction To Statistical Pattern Recognition, Academic Press, New York, 1972.
- [141. Gelsema, E.S., Kanal, L.N., Pattern Recognition In Practice, North-Holland Pub. Co., Amsterdam, Holland, 1980.
- [151. Kanal, L.N., Rosenfeld, A., Progress In Pattern Recognition, Vol. 1, North-Holland Pub. Co., 1981.
- [161. Kohonen, T., Self-Organization And Associative Memory, Springer-Verlag, Berlin, 1984.
- [171. Michalski, R.S., et.al., Machine Learning: An AI Approach, Tioga Pub. Co., Palo Alto, California, 1983.

- [191. Pavlidis, T., Structural Pattern Recognition, Springer-Verlag, Berlin, Heidelberg, New York, 1977.
- [191. Pearl, Judea, Heuristics, Addison-Wesley Publishing Co., Reading, Massachusetts, 1984.
- [201. Rich, Elaine, Artificial Intelligence, McGraw-Hill Co. New York, 1983.
- [211. Young, T.Y., Calvert, T.W., Classification, Estimation And Pattern Recognition, American Elsevier Publishing Co., New York, 1974.

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX II. Edge Detection and Image Processing Using a
Gradient and Edge Following Method

CHAPTER ONE

INTRODUCTION .

A most significant branch of recognition technology is "image processing". It involves transforming pictures into forms that facilitate analysis by machines and/or humans.

Edge detection is generally an important step in automatic image processing. For the most part, edge detection is a two-step process. The initial step consists of determining the local edges (e.g., gray level discontinuities) of an image. The second step in this process involves a method of connecting local edges into global edges. To accomplish this, we shall use an edge following technique.

This paper is an investigation of one method of edge detection. Programs are provided which illustrate the two step process involved in the problem of edge detection. Generally, the detection of edges is started by performing local operations on picture neighborhoods through the use of an edge operator. These operations, sometimes regarded as local edge operations compare intensity values within small regions of the picture. Among the many edge operators which

may be used for detecting edges, the gradient will be used in this paper. It is one of the most commonly used edge operators. The result of the gradient is a vector attached to each picture point indicating the direction of maximum gray level change. The picture is similar to the magnetic field in the space around a magnetic bar where lines of force are used to show the direction a particle would tend to move. The lines of force, like the orientations from the gradient, form a convenient way to create a visual image of the situation.

The gradient operator is applied to each pixel or element in the input picture, thus transforming the picture into an array of gradient vectors. Each gradient magnitude is compared to some threshold value. Those magnitudes less than the threshold value are, for all practical purposes, disregarded as edges. Gradient magnitudes greater than or equal to the threshold values represent significant edges in the image. The technique of thresholding is a way of segmenting or separating the object in a picture from its background, assuming that the picture is simply an object-background image. The process described above is generally referred to as a preprocessing or early processing technique.

A method for edge following will also be presented in this paper. The edge following technique computes a "goodness of fit" measurement for each pair of adjacent pixels. The goodness measure is based on the gradient of each of the two pixels. Combining pairs of pixels that fit well produces longer edges. The outcome of this method is in the form of a chain-coded representation which illustrates the direction of most change in the picture.

CHAPTER TWO

DESCRIPTION OF THE PROBLEM

This paper focuses upon two types of image processing problems. The first, referred to as local edge detection, basically involves locating discontinuities of gray levels in a given picture. The second problem, edge following, involves linking local edges to form global edges. The following paragraphs will be a discussion of these concerns.

2.1 THE EDGE DETECTION PROBLEM

Central to work involving picture processing by a computer is representation of the picture. Since the actual object cannot be manipulated in the computer, a model which can be used is generally constructed to represent the real object.

A picture is typically recorded as a vector or array of fixed dimensions. It consists of components which may be accessed arbitrarily by specifying an index which gives the position of the component within the array. For example, the picture may be given as a rectangular array of $M \times N$ dimensions, where $x=\{0\dots M\}$ and $y=\{0\dots N\}$.

Before a picture can be analyzed by the digital computer, it must be converted to a discrete form or binary image. This is usually achieved by recording the picture photographically and then transforming it based on intensity values within local regions of the picture. In the case of the black and white picture, these values are referred to as gray levels.

A point, (x,y) , in the digitized picture is referred to as a pixel. Each pixel can be said to represent a square, with the horizontal and vertical neighbors of (x,y) sharing a common boundary, and its diagonal neighbors touching it only at a corner. (See fig. 1)

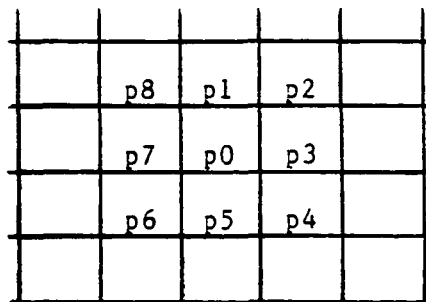


fig. 1. Representation of a pixel on a 3 x 3 grid and its eight adjacent neighbors.

Except for those pixels in the first or last row or column, each pixel in a rectangular grid representation has eight immediate neighbors associated with it as shown in fig. 1. The neighbors are identified as

$$(x,y) \text{----} \rightarrow (x-1,y-1), (x-1,y), (x-1,y+1), (x,y-1) \\ (x,y+1), (x+1,y-1), (x+1,y), (x+1,y+1).$$

An edge element (x,y) defines the common boundary between two adjacent elements. An edge is a sequence of edge elements $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$, such that the sequence of boundary segments they define on the rectangular grid is connected. The direction of a segment is determined by moving clockwise around the boundaries of the edge element.

Each pixel has a gray level value which can be modified by applying an operation to it. There should be an abrupt change (edge) in gray level at the boundary between black areas and white areas. For instance, by applying an edge detecting operator to a black and white picture, edges can be detected. A simple operator for detecting edges is the gradient. This operation on a picture computes a value (gradient magnitude) which reflects the amount of variation in gray level at a pixel. That is, high values are in areas

where edges are located and low values elsewhere. It also computes a direction which corresponds with the direction of maximum change in gray level.

Thresholding these magnitudes is a way of segmenting the picture into two regions of gray levels. Points where the gradient magnitude is greater than or equal to a certain threshold is determined to be an edge element candidate and is set to the number 1, 0's are assigned elsewhere. This new image is a two-valued image sometimes referred to as a binary image or discrete values.

2.2 THE EDGE FOLLOWING PROBLEM

Edge following is implemented after the set of edge element candidates have been isolated. The focus of the edge following technique is the location of connected edge elements and a process for linking these components as an indication of the "goodness" of an edge. Goodness here refers to a measure of how well two adjacent edge elements "fit together". A local neighborhood of a pixel is defined as the pixel and its eight adjacent neighbors on a 3 x 3 grid. A check in the direction of each adjacent neighbor for one which has the largest "goodness" in relation to the central element, yields the next element. All edge element

candidates (those pixels marked with the number 1) would be processed in the same manner. The goodness operator, a locally applied operator, computes a goodness of fit measure in the local neighborhood of each central pixel, which allows these comparisons within the neighborhood to be successfully implemented.

As aforementioned, when we view a pixel as a square on a rectangular grid, it can have eight adjacent neighbors. To determine which neighbor has been chosen to have the best fit to the central pixel in a local neighborhood, a direction code represented by the numbers {1,2,3,...8} is assigned to each of the eight neighboring positions. The coding procedure used to identify a neighbor is straightforward, it assigns the adjacent pixel vertically upward from the central pixel the code 1, and other codes can be determined by making assignments progressively while moving in a clockwise direction. An illustration of the coding scheme is provided in figure 2.

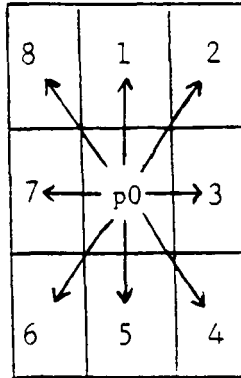


fig. 2. Coding scheme used in the assignment of direction codes.

In generally, we can say that the codes represent arrows or orientations as indicated by the coding scheme. As the picture is scanned, a direction code is assigned which corresponds with the neighbor having the "best fit" to the central pixel. The codes generated from this can be said to represent a "coded picture" when displayed in the form of a matrix, as shown in fig. 3.

0	0	0	0	0	0
0	0	3	3	7	0
0	1	0	0	7	0
0	1	0	0	7	0
0	1	3	3	7	0
0	0	0	0	0	0

Chain-codes: 3 3 7 1 7 1 7 1 3 3 7

fig. 3. The matrix of chaincodes.

Here, the coded picture illustrates the direction in which the maximum amount of change is occurring in the picture.

CHAPTER THREE

IMPLEMENTATION NOTES

The thrust of this study is edge detection and edge following. The proceeding is an explanation of several programming considerations. The program was implemented in standard PASCAL on the VAX 11/780.

3.1 INPUT DATA

The picture will be in the form of a black and white representation. Each pixel in the picture provide some detail about the picture. In this case, a description is needed which represents both the image (black area) and the background (white area) of the picture. By using an asterisk (*) at each pixel where the image is located, representation of the desired image is generated. The background is made up of any other character. Using this arrangement, a number of different images may be created and processed, respectively. This picture will represent the original or input picture, as illustrated in fig. 4.


```

b  b  b  b  b  b
b  *  *  *  *  b
b  *  b  b  *  b
b  *  b  b  *  b
b  *  *  *  *  b
b  b  b  b  b  b

```

fig. 4. A representation of the input picture.

The picture is placed into a data file which is a way of assuring that it will not be destroyed or altered in any way. However, for the purpose of analysis, the picture is copied into an array, of a size equal to the data file.

3.2 INTERNAL CODE

In order to further process the picture it is necessary to encode the input picture numerically. For this purpose, descriptive values are assigned to each pixel of the picture. This is accomplished by scanning the picture from left to right, top to bottom, whenever an asterisk is encountered the number 10 is assigned to that pixel, otherwise a 0 is assigned. The values are descriptive of

the data located at each pixel. In this case, the values represent the two regions of gray levels. The number 10 represents the black area and 0 is for the white area. The choice of the numbers is arbitrary, but must be different enough to distinguish the two regions. This numerical description of the picture is stored in the new array, NUMPIC[x,y]. It is obvious that before the picture is completely analyzed, it will undergo a number of transformations. In fact NUMPIC[x,y] will be used in subsequent algorithms for further manipulation and gray level interpretations. Using the input picture in fig. 4, the data in the new array would be represented as shown in fig. 5.

```
0 0 0 0 0 0
0 10 10 10 10 0
0 10 0 0 10 0
0 10 0 0 10 0
0 10 10 10 10 0
0 0 0 0 0 0
```

fig. 5. The numerically encoded picture, NUMPIC.

CHAPTER FOUR

PROGRAM DESCRIPTION

4.1 EDGE DETECTION TECHNIQUE

It is known, in general, that an edge represents a location in the picture where an abrupt change occurs. To determine where these changes are occurring, a gradient operator is applied to the picture. Since the gradient operator is a continuous operator, it must be converted to a difference operation for use on a discrete picture.

We use :

$$\text{Diffx} = \text{Numpic}[x,y] - \text{Numpic}[x+1,y]$$

$$\text{Diffy} = \text{Numpic}[x,y] - \text{Numpic}[x,y+1]$$

Using data from the numerically encoded picture, this process would generate data similar to that shown in figs. 6a and 6b.

fig. 6.

(a) Diffx

0	-5	-5	-5	-5	0
0	5	10	10	5	0
0	5	0	0	5	0
0	5	-5	-5	5	0
0	5	0	0	5	0
0	10	10	10	10	0

(b) Diffy

0	0	0	0	0	0
-5	10	0	-5	10	0
-5	5	5	5	10	0
0	0	0	0	0	0
-5	5	5	5	10	0
-5	10	0	-5	10	0

fig. 6. (a) and (b) Represent data in the
two difference matrices, Diffx and Diffy.

Using Diffx and Diffy as components of the gradient, this operator is now represented as:

$$\text{Gradient} = \text{Sqrt}(\text{Diffx}[x,y]^2 + \text{Diffy}[x,y]^2)$$

Gradient magnitudes are generated by applying the operator to each element in the numeric picture starting in the first row and ending in the last one. (See fig. 7a) With this data, edges can be located by thresholding the magnitudes. The threshold is supplied at runtime.

The magnitudes generated from this operation which are greater than the threshold value are recorded as edge data, and are marked with the number 1, otherwise a value of 0 is assigned. The 1's and 0's are stored into an array which represents the picture as discrete values. (See fig 7b)

fig. 7.

0	5	5	5	5	0
5	7	11	11	11	0
5	11	0	5	11	0
5	11	5	7	11	0
5	11	11	11	14	0
0	0	0	0	0	0

(a) Picture of gradient magnitudes.

0	0	0	0	0	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	0	1	0
0	1	1	1	1	0
0	0	0	0	0	0

(b) Image of discrete values (local edges).

4.2 EDGE FOLLOWING METHOD

Given the local edges found in the proceeding operation, we use the goodness operator to determine how well adjacent pixels are connected to the central pixel. The goodness operator uses gradient magnitudes and information from the two differenced matrices, (Diffx[x,y],Diffy[x,y]). The goodness operator is defined as:

$$\text{GOODN} = \frac{((\text{Diffx}[x,y] * \text{Diffx}[x_n, y_n]) + (\text{Diffy}[x,y] * \text{Diffy}[x_n, y_n]))}{(\text{Grad}[x_n, y_n])}$$

In this formula, (x_n, y_n) locates data at the position of an adjacent neighbor and (x, y) locates data at the central pixel in a local neighborhood. Using this equation, a goodness measure is computed for each of the neighbors in the local neighborhood of the central pixel. The goodness value for the eight neighbors are compared to each other. The "best goodness" is recorded together with the position of the adjacent pixel with the best fit, the position of the central pixel, and its gradient magnitude.

Recall, the gradient computes an orientation of discontinuity, which are basically directed lines indicating the direction of maximum change in the picture. By knowing the orientation of the line formed from the x and y coordinates of the central pixel in terms of its gradient ($\text{Diff}_x, \text{Diff}_y$) and that of its adjacent neighbor of best fit within the local neighborhood, an angle can be formed. The angle is formed when an adjacent point is projected onto the central point. The two vectors can be used to reflect how parallel or perpendicular the gradients of two pixels are to each other. The more parallel the gradients are, the smaller the goodness measure. The more perpendicular the gradients are, the larger the goodness measure. More parallel gradients indicate a more likely (smoother) continuation of an edge.

Each of the eight neighbors in the region is assigned a code $\{1, 2, \dots, 8\}$ moving in a clockwise fashion. The code represents the direction in which the edge is moving. It is also recorded as data from this procedure. Here the edge pixel represents the central pixel in the local neighborhood and the neighbor with the best goodness reflects the adjacent neighbor with a magnitude most related to the magnitude of the central pixel. Local edges are generated

by applying the goodness operator to each local neighborhood where a pixel at an edge exists.

The edge following method is controlled by a procedure which produces a record of local edges, a gradient magnitude, a goodness measure, and a chain-coded direction. An illustration of a chain-coded edge image for our example is shown in fig. 8.

0	0	0	0	0	0
0	0	3	7	1	0
0	1	0	0	1	0
0	5	0	0	5	0
0	3	3	7	5	0
0	0	0	0	0	0

fig. 8. The chain-coded image for sample input picture.

In this way, the chain codes can be used to link the local edges together to form a global edge.

CHAPTER FIVE

PROGRAMMING STRATEGY

A method for edge detection is discussed in this section which allows for simple patterns to be conveniently represented and analyzed for image processing. The strategy used in solving the edge detection/edge following problem is outlined below.

5.1 MAJOR DATA STRUCTURES

RSIZE : row size

CSIZE : column size

UBORDERX and LBORDERX : upper and lower
boundaries of the x
coordinate

UBORDERY and LBORDERY : upper and lower
boundaries of the y
coordinate

DIFFX and DIFFY : components of the gradient

GRADIENT : length of the gradient

THRESHOLD : a value initiated at run time

CTX,CTY : the central pixel of the local
neighborhood

ORIGPIC[ctx,cty] : a storage location for the
picture

NUMPIC[ctx,cty] : an array holding the numeric
picture

ADJPTARY : an array of adjacent neighbors

ADJX,ADJY : the adjacent pixel with the
best fit

GOODN : a measure of how well an adjacent pixel
fits together with a central pixel

BESTGOODN : the goodness measure of the
adjacent pixel with the best
fit to the central pixel

CODE[ctx,cty] : array of direction codes

RESULT[ctx,cty] : array of 1's and 0's

5.2 INPUT FILE DESCRIPTION

PICTURE is a text file where the input picture is stored. Using asterisks (*) to outline the image (black area) and any other character to represent the background (white area), the picture is created. When a new picture is

to be processed, the existing input picture would need to be discarded in order to create a new one in the file.

5.3 MAJOR PROCEDURES

A. MAIN PROGRAM

1. Read the picture into array ORIGPIC.
(PICMATRX)
2. Encode the picture numerically, by scanning the picture and replacing the asterisks encountered with the number 10, otherwise assign 0. Store the new image in array NUMPIC. (CHANGEPICT)
3. Develop the components (DIFFx and DIFFy) of the gradient operator. (DIFFERENCE)
4. Apply the gradient operator to the picture.
Compare the gradient magnitudes to the threshold; if the gradient[ctx,cty] is greater than or equal to the threshold assign a 1 to pixel, otherwise assign a value of 0.
Store this data in array RESULT. (GRADTHRSHTD)
5. Search and produce the chaincodes. Store the codes into array CODE. (NEXTSEARCH)
6. Display array CODE. (FINALMTRX).

B. NEXTSEARCH

For each pixel with a gradient magnitude greater than the threshold do the following:

1. Find the adjacent neighbors' of the pixel and store this data in array ADJPTARY.
(FINDADJACENT)
2. Compute a goodness of fit measure for each adjacent neighbor in the local neighborhood of the central pixel. Compare the values within the local neighborhood to determine which has the largest "goodness" relative the central pixel. Output the coordinates of the central pixel, the coordinates of the adjacent neighbor with the best goodness, the gradient magnitude of the central pixel and the goodness measure. (GOODNESS)
3. Direction code is computed and stored in array CODE.

C. FINALMTRX

Output the array of direction codes.

CHAPTER SIX

CONCLUSION

This paper provides a system of picture processing for the detection of edges in gray level pictures. To illustrate the effectiveness of program EDGEDETECT several images have been processed. Images tested in this study are size 10 x 10. In this program, the user must create the input picture in a text file before the processing begins and must also supply a threshold value at run time. By entering the threshold interactively, it has been determined that for the examples used, threshold values between 8 and 14 are considered appropriate. That is, any number greater than 14 will cause a loss of relevant edge data and values less than 8 will pickup undersirable data.

An edge following technique is also presented which illustrates the concept of edge "goodness". The edge following technique uses a 3x3 grid representation, which includes a central pixel and its eight immediate neighbors. Using a local goodness operator in local neighborhoods of the picture, yields a lists of connected elements. A coded direction scheme is employed which plots these relationships

in a chain-coded matrix. The coded picture may be interpreted as a direction matrix where the codes represent orientations of directionality.

Basically, the mechanisms involved in the edge detection/edge following technique include a series of transformations of an input picture. Starting with an original image, a new image is produced. In a sense, our beforehand knowledge of the original picture is limited, we know more about the new image since each transformation provides additional information which can be better interpreted by us. For example, the comparison between the original picture (figs. 4 and 5) and the chain-coded image (fig. 8) produces knowledge about the original picture that we can better understand. In the instance of our example, we can interpret the codes to present the direction of maximum change in the picture.

One of the difficulties with the program described in its present form is the problem it has in interpreting the corners of an object during the scanning of the pictures to locate local edges. There was also a limitation in the types of pictures which could be created. Simple patterns were primarily used in this investigation.

Despite these drawbacks, the pictures sampled indicated that program EDGEDETCT was effective in the extraction of local edges. In addition, the program was capable of growing an edge and determining the shape of objects.

Although the program sends the output data to a standard data file, it can also be captured in the Virtual Memory Storage (VMS) using a Define/User command. A copy of the program is provided in the APPENDIX section of the paper, along with printouts of the sampled data.

A P P E N D I X

ORIGINAL PAGE IS
OF POOR QUALITY

```
PROGRAM EDGEDETECT(INPUT,OUTPUT,PICTURE);
const RSIZE=9;
      CSIZE=9;
      UBORDERX=9;
      LBORDERX=0;
      UBORDERY=9;
      LBORDERY=0;

type
  GRADARRAY=ARRAY[0..CSIZE,0..RSIZE]OF REAL;
  ORIGINAL=ARRAY[0..CSIZE,0..RSIZE]OF CHAR;
  ORIG=ARRAY[0..CSIZE,0..RSIZE]OF REAL;
  DELTAY=ARRAY[0..CSIZE,0..RSIZE]OF REAL;
  DELTAX=ARRAY[0..CSIZE,0..RSIZE]OF REAL;
  TEMPTYPE=ARRAY[0..CSIZE,0..RSIZE]OF INTEGER;
  DIRECTION=ARRAY[0..CSIZE,0..RSIZE]OF INTEGER;
  SODDVALS=ARRAY[0..7]OF REAL;
  POSITION=RECORD
    X: INTEGER;
    Y: INTEGER;
END;
ADJARRAY=ARRAY[0..7]OF POSITION;

var PICTURE:TEXT;
    ORIGPIC:ORIGINAL;
    NUMPIC:ORIG;
    DIFFX:DELTAX;
    DIFFY:DELTAY;
    ADJPTARY:ADJARRAY;
    GRAD:GRADARRAY;
    RESULT,TEMP:TEMPTYPE;
    CTX,CTY:INTEGER;
    THRESHOLD:REAL;
    BESTGOODN:REAL;
    PX,PY:INTEGER;
    CODE:DIRECTION;
```

**ORIGINAL PAGE IS
OF POOR QUALITY**

```
*****  
* This procedure copies the picture of asterisks and blanks from the *  
* data file PICTURE.dat where it is stored, into matrix array ORISPIC(x,y) *  
*****
```

```
PROCEDURE PICMATX(ORIGPIC:ORIGINAL);  
VAR X,Y:INTEGER;  
  
BEGIN (** PICMATX **)  
  RESET(PICTURE);  
  WRITELN;  
  BEGIN  
    FOR X:=0 TO RSIZE DO  
      BEGIN  
        FOR Y:=0 TO CSIZE DO  
          BEGIN  
            READ(PICTURE,ORIGPIC(X,Y));  
          END;  
          READLN(PICTURE);  
        END;  
        WRITELN;  
        WRITELN;  
      END;  
    END;  
  END;  
END; ( END OF PICMATX )
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
*****  
* Procedure CHANGEPIC, scans the picture and assigns a number to each *  
* point in the picture. The number 10 is assigned to points where an *  
* asterisk resides and 0's elsewhere. This new picture is stored in *  
* array NUMPIC.[x,y]. *  
*****
```

```
PROCEDURE CHANGEPIC(ORIGPIC:ORIGINAL;VAR NUMPIC:ORIG);  
VAR X,Y:INTEGER;
```

```
BEGIN (** CHANGEPIC **)  
  WRITELN(' THIS IS THE PICTURE');  
  WRITELN;  
  FOR X:=0 TO RSIZE DO  
    BEGIN  
      FOR Y:=0 TO CSIZE DO  
        BEGIN  
          IF ORIGPIC[X,Y]= '*'  
            THEN NUMPIC[X,Y]:=10  
            ELSE  
              NUMPIC[X,Y]:=0;  
          WRITE(TRUNC(NUMPIC[X,Y]):4);  
        END;  
        WRITELN;  
        WRITELN;  
      END;  
    WRITELN;  
  END; { END OF CHANGEPIC }
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
*****  
* This procedure divides the picture into two difference matrices, *  
* the vertical difference (DIFFX(x,y)) and the horizontal difference *  
* (DIFFY(x,y)). This difference information is the approximation *  
* to the gradient used in the program. *  
*****
```

```
PROCEDURE DIFFERENCE(NUMPIC:ORIG;VAR DIFFX:DELTAX;VAR DIFFY:DELTAY);  
VAR X,Y:INTEGER;
```

```
BEGIN (** DIFFERENCE **)  
  WRITELN(' DIFF(Y)');  
  FOR X:=0 TO RSIZE DO  
    BEGIN  
      WRITELN:  
        FOR Y:=0 TO CSIZE DO  
          BEGIN  
            DIFFY(X,Y):=(NUMPIC(X,Y))-(NUMPIC(X,Y+1))/2;  
            WRITE(TRUNC(DIFFY(X,Y):4);  
          END;  
          WRITELN;  
          WRITELN:  
        END;  
      WRITELN;  
      WRITELN;  
      WRITELN(' DIFF(X)');  
      WRITELN:  
        FOR X:= 0 TO (RSIZE - 1) DO  
          BEGIN  
            WRITELN;  
            FOR Y:=0 TO CSIZE DO  
              BEGIN  
                DIFFX(X,Y):=(NUMPIC(X,Y))-(NUMPIC(X+1,Y))/2;  
                WRITE(TRUNC(DIFFX(X,Y):4);  
              END;  
              WRITELN;  
              WRITELN;  
            END;  
          WRITELN;  
          WRITELN;  
        END;  
      WRITELN;  
      WRITELN;  
      WRITELN(' END OF DIFFERENCE ');
```

ORIGINAL PAGE IS
OF POOR QUALITY

```

*****
* Procedure GRADTHRSHD computes a gradient magnitude at edge pixel which *
* is compared to a threshold value that the user designates. Points where *
* the gradient magnitude is greater than the threshold are marked with the *
* the number 1, otherwise a 0 is assigned. The 1's and 0' are stored *
* in array RESULT[ctx,cty]. *
*****

```

```

PROCEDURE GRADTHRSHD(CTX,CTY:INTEGER;VAR RESULT:TEMPTYPE;
VAR GRAD:GRADARRAY);

```

```

VAR THRESHOLD:REAL;

```

```

BEGIN

```

```

    WRITELN;

```

```

    WRITELN('PLEASE ENTER THE THRESHOLD VALUE ');

```

```

    READLN(THRESHOLD);

```

```

    FOR CTX:=0 TO RSIZE DO

```

```

        FOR CTY:=0 TO CSIZE DO

```

```

            GRAD[CTX,CTY]:=SQRT(((DIFFX[CTX,CTY])*(DIFFX[CTX,CTY]) +
            ((DIFFY[CTX,CTY])*(DIFFY[CTX,CTY])));

```

```

        FOR CTX:=0 TO RSIZE DO

```

```

            FOR CTY:= 0 TO CSIZE DO

```

```

                BEGIN

```

```

                    IF GRAD[CTX,CTY] >= THRESHOLD THEN

```

```

                        RESULT[CTX,CTY]:=1

```

```

                    ELSE

```

```

                        RESULT[CTX,CTY]:=0;

```

```

                END;

```

```

    WRITELN;

```

```

    WRITELN;

```

```

    WRITELN;

```

```

    WRITELN(' GRAD RESULT:');

```

```

    WRITELN;

```

```

    FOR CTX:=0 TO RSIZE DO

```

```

        BEGIN

```

```

            FOR CTY:=0 TO CSIZE DO

```

```

                WRITE(RESULT[CTX,CTY]:4);

```

```

                WRITELN;

```

```

            END;

```

```

    WRITELN;

```

```

    WRITELN;

```

```

    END; (END OF GRADTHRSHD)

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
#####  
: This procedure locates points adjacent to each of the pixels in the :  
: picture and checks for the border of the picture. :  
: :  
#####
```

```
PROCEDURE FINDADJACENT(CTX,CTY:INTEGER;VAR ADJTARY:ADJARRAY);  
VAR I:INTEGER;
```

```
BEGIN (## FINDADJACENT ##)
```

```
FOR I:=0 TO 7 DO
```

```
  BEGIN
```

```
    ADJTARY[I].X:=-99;
```

```
    ADJTARY[I].Y:=-99;
```

```
  END;
```

```
IF CTY=LBORDERX
```

```
  THEN IF CTX=LBORDERX
```

```
    THEN BEGIN
```

```
      ADJTARY[0].X:=CTX+1;
```

```
      ADJTARY[0].Y:=CTY;
```

```
      ADJTARY[1].X:=CTX+1;
```

```
      ADJTARY[1].Y:=CTY+1;
```

```
      ADJTARY[2].X:=CTX;
```

```
      ADJTARY[2].Y:=CTY+1;
```

```
    END
```

```
  ELSE IF CTX=UBORDERX
```

```
    THEN BEGIN
```

```
      ADJTARY[2].X:=CTX;
```

```
      ADJTARY[2].Y:=CTY+1;
```

```
      ADJTARY[3].X:=CTX-1;
```

```
      ADJTARY[3].Y:=CTY+1;
```

```
      ADJTARY[4].X:=CTX-1;
```

```
      ADJTARY[4].Y:=CTY;
```

```
    END
```

```
  ELSE BEGIN
```

```
    ADJTARY[0].X:=CTX+1;
```

```
    ADJTARY[0].Y:=CTY;
```

```
    ADJTARY[1].X:=CTX+1;
```

```
    ADJTARY[1].Y:=CTY+1;
```

```
    ADJTARY[2].X:=CTX;
```

```
    ADJTARY[2].Y:=CTY+1;
```

```
    ADJTARY[3].X:=CTX-1;
```

```
    ADJTARY[3].Y:=CTY+1;
```

```
    ADJTARY[4].X:=CTX-1;
```

```
    ADJTARY[4].Y:=CTY;
```

```
  END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
ELSE IF CTY=UBORDEPY
  THEN IF CTX=LBORDERX
    THEN BEGIN
      ADJPTARY[0].X:=CTX+1;
      ADJPTARY[0].Y:=CTY;
      ADJPTARY[6].X:=CTX;
      ADJPTARY[6].Y:=CTY-1;
      ADJPTARY[7].X:=CTX+1;
      ADJPTARY[7].Y:=CTY-1;
    END
  ELSE IF CTX=UBORDERX
    THEN BEGIN
      ADJPTARY[4].X:=CTX-1;
      ADJPTARY[4].Y:=CTY;
      ADJPTARY[5].X:=CTX-1;
      ADJPTARY[5].Y:=CTY-1;
      ADJPTARY[6].X:=CTX;
      ADJPTARY[6].Y:=CTY-1;
    END
  ELSE BEGIN
      ADJPTARY[4].X:=CTX-1;
      ADJPTARY[4].Y:=CTY;
      ADJPTARY[5].X:=CTX-1;
      ADJPTARY[5].Y:=CTY-1;
      ADJPTARY[6].X:=CTX;
      ADJPTARY[6].Y:=CTY-1;
      ADJPTARY[7].X:=CTX+1;
      ADJPTARY[7].Y:=CTY-1;
    END
  ELSE IF CTX=LBORDERX
    THEN BEGIN
      ADJPTARY[0].X:=CTX+1;
      ADJPTARY[0].Y:=CTY;
      ADJPTARY[1].X:=CTX+1;
      ADJPTARY[1].Y:=CTY+1;
      ADJPTARY[2].X:=CTX;
      ADJPTARY[2].Y:=CTY+1;
      ADJPTARY[6].X:=CTX;
      ADJPTARY[6].Y:=CTY-1;
      ADJPTARY[7].X:=CTX+1;
      ADJPTARY[7].Y:=CTY-1;
    END
```


ORIGINAL PAGE IS
OF POOR QUALITY

```
ELSE IF CTX=UBORDERX
  THEN BEGIN
    ADJPTARY[2].X:=CTX;
    ADJPTARY[2].Y:=CTY+1;
    ADJPTARY[3].X:=CTX-1;
    ADJPTARY[3].Y:=CTY+1;
    ADJPTARY[4].X:=CTX-1;
    ADJPTARY[4].Y:=CTY;
    ADJPTARY[5].X:=CTX-1;
    ADJPTARY[5].Y:=CTY-1;
    ADJPTARY[6].X:=CTX;
    ADJPTARY[6].Y:=CTY-1;
    END
  ELSE BEGIN
    ADJPTARY[0].X:=CTX+1;
    ADJPTARY[0].Y:=CTY;
    ADJPTARY[1].X:=CTX+1;
    ADJPTARY[1].Y:=CTY+1;
    ADJPTARY[2].X:=CTX;
    ADJPTARY[2].Y:=CTY+1;
    ADJPTARY[3].X:=CTX-1;
    ADJPTARY[3].Y:=CTY+1;
    ADJPTARY[4].X:=CTX-1;
    ADJPTARY[4].Y:=CTY;
    ADJPTARY[5].X:=CTX-1;
    ADJPTARY[5].Y:=CTY-1;
    ADJPTARY[6].X:=CTX;
    ADJPTARY[6].Y:=CTY-1;
    ADJPTARY[7].X:=CTX+1;
    ADJPTARY[7].Y:=CTY-1;
  END; (END OF IF)
  WRITELN;
  (WRITELN('POSITION', CTX,CTY);)
  WRITELN;
  ( FOR I:=0 TO 7 DO
    WRITELN(ADJPTARY[I].X,ADJPTARY[I].Y);)
  END; ( END OF FINDADJACENT )
```

ORIGINAL PAGE IS
OF POOR QUALITY

```

*****
* The following procedure is used to compute a goodness of fit measure *
* for each pair of adjacent pixels. This measure determines whether *
* an adjacent neighbor to a central pixel fits well together. *
*
*****

```

```

PROCEDURE GOODNESS(CTX,CTY: INTEGER;ADJPTARY:ADJARRAY;VAR BESTGOODN:REAL;
VAR ADJX,ADJY: INTEGER;GRAD:GRADARRAY;CODE: DIRECTION);

```

```

VAR I,K: INTEGER;
GOODN: REAL;

```

```

BEGIN (** GOODNESS **)

```

```

BESTGOODN:=-999;

```

```

ADJX:=0;

```

```

ADJY:=0;

```

```

FOR I:=0 TO 7 DO

```

```

IF (ADJPTARY[I].X)=0) THEN

```

```

BEGIN

```

```

IF (GRAD[ADJPTARY[I].X,ADJPTARY[I].Y])=0 THEN

```

```

GOODN:=0

```

```

ELSE

```

```

GOODN:=(ABS((DIFFX[CTX,CTY]*DIFFX[ADJPTARY[I].X,ADJPTARY[I].Y])+
(DIFFY[CTX,CTY]*DIFFY[ADJPTARY[I].X,ADJPTARY[I].Y]))/
(GRAD[ADJPTARY[I].X,ADJPTARY[I].Y]));

```

```

IF BESTGOODN<GOODN

```

```

THEN BEGIN

```

```

BESTGOODN:=GOODN;

```

```

ADJX:=ADJPTARY[I].X;

```

```

ADJY:=ADJPTARY[I].Y;

```

```

END;

```

```

END;

```

```

WRITELN;

```

```

WRITELN(CTX,CTY:2,(TRUNC(GRAD[CTX,CTY])),ADJX,ADJY:2,(TRUNC(BESTGOODN)));

```

```

WRITELN;

```

```

WRITELN;

```

```

END: (END OF GOODNESS)

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
*****  
* Procedure NEXTSEARCH calls procedures FINDADJACENT and GOODNESS, then *  
* assigns a direction code which corresponds to each respective *  
* adjacent point. The code is generated in this procedure in a *  
* clockwise direction. The codes are stored in array CODE[ctx,cty]. *  
*****
```

```
PROCEDURE NEXTSEARCH (CTX,CTY:INTEGER;VAR CODE:DIRECTION);  
BEGIN (** NEXTSEARCH **)  
  FINDADJACENT(CTX,CTY,ADJPTARY);  
  GOODNESS(CTX,CTY,ADJPTARY,BESTGOODN,ADJX,ADJY,GRAD,CODE);  
  BEGIN  
    IF (ADJX=ADJPTARY[0].X) AND (ADJY=ADJPTARY[0].Y) THEN CODE[CTX,CTY]:=1;  
    IF (ADJX=ADJPTARY[1].X) AND (ADJY=ADJPTARY[1].Y) THEN CODE[CTX,CTY]:=2;  
    IF (ADJX=ADJPTARY[2].X) AND (ADJY=ADJPTARY[2].Y) THEN CODE[CTX,CTY]:=3;  
    IF (ADJX=ADJPTARY[3].X) AND (ADJY=ADJPTARY[3].Y) THEN CODE[CTX,CTY]:=4;  
    IF (ADJX=ADJPTARY[4].X) AND (ADJY=ADJPTARY[4].Y) THEN CODE[CTX,CTY]:=5;  
    IF (ADJX=ADJPTARY[5].X) AND (ADJY=ADJPTARY[5].Y) THEN CODE[CTX,CTY]:=6;  
    IF (ADJX=ADJPTARY[6].X) AND (ADJY=ADJPTARY[6].Y) THEN CODE[CTX,CTY]:=7;  
    IF (ADJX=ADJPTARY[7].X) AND (ADJY=ADJPTARY[7].Y) THEN CODE[CTX,CTY]:=8;  
    WRITELN('DIRECTION CODE:',CODE[CTX,CTY]:8);  
  END;  
END; ( END OF NEXTSEARCH )
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
*****  
* This procedure outputs the matrix of direction codes, CODE[X,Y]. *  
*****
```

```
PROCEDURE FINALMTRX(TEMP:TEMPTYPE;VAR CODE:DIRECTION);  
  VAR X,Y:INTEGER;  
  
  BEGIN (** FINALMTRX **)  
    WRITELN;  
    WRITELN(' DIRECTION CODE MATRIX:');  
    WRITELN;  
    FOR X:=0 TO RSIZE DO  
      BEGIN  
        FOR Y:=0 TO CSIZE DO  
          WRITE(CODE[X,Y]:4);  
          WRITELN;  
        END;  
      WRITELN;  
    END; ( END OF FINALMTRX )
```

```

BEGIN (** MAIN PROGRAM **)

PICMATX(ORIGPIC);
CHANGEPICT(ORIGPIC, NUMPIC);
DIFFERENCE(NUMPIC, PICX, PICY);
GRADTHRESHLD(CTX, CTY, RESULT, GRAD);

WRITELN('START POSITION', '    IGRAD', '    ADJ-POSITION', '    BEST SNESS');
FOR CTX:=0 TO RSIZE DO
  FOR CTY:=0 TO CSIZE DO
    IF RESULT[CTX,CTY]=1
      THEN BEGIN
        NEXTSEARCH(CTX,CTY,CODE);
        END;
        WRITELN;
        WRITELN;
FINALMTRX(TEMP, CODE);
END. ( END OF MAIN )

```

ORIGINAL PAGE IS
OF POOR QUALITY

THE NUMERICALLY ENCODED PICTURE :

- The block letter C

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	10	10	10	10	10	10	0	0
0	0	10	0	0	0	0	0	0	0
0	0	10	0	0	0	0	0	0	0
0	0	10	0	0	0	0	0	0	0
0	0	10	0	0	0	0	0	0	0
0	0	10	10	10	10	10	10	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(Y)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	-5	5	5	5	5	5	10	0	0
0	-5	10	0	0	0	0	0	0	0
0	-5	10	0	0	0	0	0	0	0
0	-5	10	0	0	0	0	0	0	0
0	-5	10	0	0	0	0	0	0	0
0	-5	5	5	5	5	5	10	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(X)

0	0	0	0	0	0	0	0	0	0
0	0	-5	-5	-5	-5	-5	-5	0	0
0	0	5	10	10	10	10	10	0	0
0	0	5	0	0	0	0	0	0	0
0	0	5	0	0	0	0	0	0	0
0	0	5	0	0	0	0	0	0	0
0	0	5	-5	-5	-5	-5	-5	0	0
0	0	10	10	10	10	10	10	0	0
0	0	0	0	0	0	0	0	0	0

GRAD/THRESHOLD RESULT:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0
0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

CHAIN-CODED IMAGE:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	7	7	0	0
0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	5	0	0	0	0	0	0	0
0	0	3	3	3	3	7	7	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

THE NUMERICALLY ENCODED PICTURE:

- The letter X

0	0	0	0	0	0	0	0	0	0
0	10	0	0	0	0	0	0	10	0
0	0	10	0	0	0	0	10	0	0
0	0	0	10	0	0	10	0	0	0
0	0	0	0	10	10	0	0	0	0
0	0	0	10	0	0	10	0	0	0
0	0	10	0	0	0	0	10	0	0
0	10	0	0	0	0	0	0	10	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(Y)

0	0	0	0	0	0	0	0	0	0
-5	10	0	0	0	0	0	-5	10	0
0	-5	10	0	0	0	-5	10	0	0
0	0	-5	10	0	-5	10	0	0	0
0	0	0	-5	5	10	0	0	0	0
0	0	-5	10	0	-5	10	0	0	0
0	-5	10	0	0	0	-5	10	0	0
-5	10	0	0	0	0	0	-5	10	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(X)

0	-5	0	0	0	0	0	0	-5	0
0	10	-5	0	0	0	0	-5	10	0
0	0	10	-5	0	0	-5	10	0	0
0	0	0	10	-5	-5	10	0	0	0
0	0	0	-5	10	10	-5	0	0	0
0	0	-5	10	0	0	10	-5	0	0
0	-5	10	0	0	0	0	10	-5	0
0	10	0	0	0	0	0	0	10	0
0	0	0	0	0	0	0	0	0	0

GRAD/THRESHOLD RESULT:

0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	1	0	0
0	0	0	1	0	0	1	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	1	0	0	0
0	0	1	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

CHAIN-CODED IMAGE :

0	0	0	0	0	0	0	0	0	0
0	2	0	0	0	0	0	0	8	0
0	0	2	0	0	0	0	4	0	0
0	0	0	6	0	0	4	0	0	0
0	0	0	0	3	2	0	0	0	0
0	0	0	8	0	0	2	0	0	0
0	0	4	0	0	0	0	2	0	0
0	4	0	0	0	0	0	0	6	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

THE NUMERICALLY ENCODED PICTURE :

- The number 8 in block form.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	10	10	10	10	10	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	10	10	10	10	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	10	10	10	10	10	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(Y)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	-5	5	5	5	5	5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	5	5	5	5	5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	5	5	5	5	5	10	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(X)

0	0	0	0	0	0	0	0	0	0
0	0	-5	-5	-5	-5	-5	-5	0	0
0	0	5	10	10	10	10	5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	5	-5	-5	-5	-5	5	0	0
0	0	5	10	10	10	10	5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	5	-5	-5	-5	-5	5	0	0
0	0	10	10	10	10	10	10	0	0

GRAD/THRESHOLD RESULT:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

CHAIN-CODED IMAGE:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	7	7	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	3	3	3	7	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	5	0	0	0	0	5	0	0
0	0	3	3	3	3	7	7	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

THE NUMERICALLY ENCODED PICTURE :

- The block letter A.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	10	10	10	10	10	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	10	10	10	10	0	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(Y)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	-5	5	5	5	5	5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	5	5	5	5	10	0	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(X)

0	0	0	0	0	0	0	0	0	0
0	0	-5	-5	-5	-5	-5	-5	0	0
0	0	5	10	10	10	10	5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	5	-5	-5	-5	-5	10	0	0
0	0	5	10	10	10	10	-5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	10	0	0	0	0	10	0	0

GRAD/THRESHOLD RESULT:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0

CHAIN-CODED IMAGE :

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	7	1	0	0
0	0	1	0	0	0	0	5	0	0
0	0	5	0	0	0	0	8	0	0
0	0	0	3	3	7	4	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	5	0	0	0	0	5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	0	0	0	0	0	0	0	0

THE NUMERICALLY ENCODED PICTURE :

-The number 8 with squared corners.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	10	10	10	10	0	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	0	10	10	10	10	0	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	0	10	10	10	10	0	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(Y)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	-5	5	5	5	10	0	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	0	-5	5	5	5	10	0	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	0	-5	5	5	5	10	0	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(X)

0	0	0	0	0	0	0	0	0	0
0	0	0	-5	-5	-5	-5	0	0	0
0	0	-5	10	10	10	10	-5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	10	-5	-5	-5	-5	10	0	0
0	0	-5	10	10	10	10	-5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	10	-5	-5	-5	-5	10	0	0
0	0	0	10	10	10	10	0	0	0

GRAD/THRESHOLD RESULT:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0

CHAIN-CODED IMAGE :

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	3	3	7	2	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	2	0	0	0	0	8	0	0
0	0	0	3	3	7	4	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	2	0	0	0	0	8	0	0
0	0	0	3	3	7	4	0	0	0
0	0	0	0	0	0	0	0	0	0

THE NUMERICALLY ENCODED PICTURE :

-The letter B.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	10	10	10	10	10	0	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	10	10	10	10	0	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	10	10	10	10	0	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(Y)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	-5	5	5	5	5	10	0	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	5	5	5	5	10	0	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	5	5	5	5	10	0	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(X)

0	0	0	0	0	0	0	0	0	0
0	0	-5	-5	-5	-5	-5	0	0	0
0	0	5	10	10	10	10	-5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	5	-5	-5	-5	-5	10	0	0
0	0	5	10	10	10	10	-5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	5	-5	-5	-5	-5	10	0	0
0	0	10	10	10	10	10	0	0	0

GRAD/THRESHOLD RESULT:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0

CHAIN-CODED IMAGE:

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	3	3	7	2	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	5	0	0	0	0	8	0	0
0	0	0	3	3	7	4	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	5	0	0	0	0	8	0	0
0	0	3	3	3	7	4	0	0	0
0	0	0	0	0	0	0	0	0	0

THE NUMERICALLY ENCODED PICTURE :

- The letter A.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	10	10	0	0	0	0
0	0	0	10	0	0	10	0	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	10	10	10	10	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	10	0	0	0	0	10	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(Y)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	-5	5	10	0	0	0	0
0	0	-5	10	0	-5	10	0	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	5	5	5	5	5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	-5	10	0	0	0	-5	10	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(X)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	-5	-5	0	0	0	0
0	0	-5	10	0	0	10	-5	0	0
0	0	5	-5	-5	-5	-5	5	0	0
0	0	5	10	10	10	10	5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	5	0	0	0	0	5	0	0
0	0	5	0	0	0	0	5	0	0

GRAD/THRESHOLD RESULT :

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	1	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	1	1	1	1	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0

CHAIN-CODED IMAGE :

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	3	2	0	0	0	0
0	0	0	4	0	0	6	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	3	3	3	7	1	0	0
0	0	1	0	0	0	0	1	0	0
0	0	5	0	0	0	0	1	0	0
0	0	5	0	0	0	0	5	0	0
0	0	0	0	0	0	0	0	0	0

THE NUMERICALLY CODED PICTURE :

- A diamond shaped figure.

0	0	0	0	0	0	0	0	0	0
0	0	0	0	10	10	0	0	0	0
0	0	0	10	0	0	10	0	0	0
0	0	10	0	0	0	0	10	0	0
0	10	0	0	0	0	0	0	10	0
0	0	10	0	0	0	0	10	0	0
0	0	0	10	0	0	10	0	0	0
0	0	0	0	10	10	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(Y)

0	0	0	0	0	0	0	0	0	0
0	0	0	-5	5	10	0	0	0	0
0	0	-5	10	0	-5	10	0	0	0
0	-5	10	0	0	0	-5	10	0	0
-5	10	0	0	0	0	0	-5	10	0
0	-5	10	0	0	0	-5	10	0	0
0	0	-5	10	0	-5	10	0	0	0
0	0	0	-5	5	10	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

DIFF(X)

0	0	0	0	-5	-5	0	0	0	0
0	0	0	-5	10	10	-5	0	0	0
0	0	-5	10	0	0	10	-5	0	0
0	-5	10	0	0	0	0	10	-5	0
0	10	-5	0	0	0	0	-5	10	0
0	0	10	-5	0	0	-5	10	0	0
0	0	0	10	-5	-5	10	0	0	0
0	0	0	0	10	10	0	0	0	0
0	0	0	0	0	0	0	0	0	0

GRAD/THRESHOLD RESULT :

0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	1	0	0	0
0	0	1	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	1	0	0
0	0	0	1	0	0	1	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

CHAIN-CODE IMAGE :

0	0	0	0	0	0	0	0	0	0
0	0	0	0	3	2	0	0	0	0
0	0	0	8	0	0	2	0	0	0
0	0	4	0	0	0	0	2	0	0
0	2	0	0	0	0	0	0	6	0
0	0	2	0	0	0	0	4	0	0
0	0	0	6	0	0	4	0	0	0
0	0	0	0	3	4	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

REFERENCES

1. Ballard, D. H. and Brown, C. M., Computer Vision, Prentice Hall, New Jersey, 1982.
2. Griffith, A. K., Edge detection in simple scenes using a priori information, IEEE Transactions On Computers, vol. C-22, 1973, 371-381.
3. Hart, P., Nilson, N. and Raphael, B., A formal basis for the heuristic determination of minimum cost paths, IEEE Transactions On Systems Science and Cybernetics, vol. 4, 1968, 100-107.
4. Marr, D., Early processing of visual information, MIT AI Memo 340, 1975.
5. Pratt, W. K., Digital Image Processing, Wiley, New York, 1978.
6. Rosenfeld, A., Picture Processing by Computer, Academic Press, New York, 1976.
7. Rosenfeld, A. and Thurston, M., Edge and curve detection for visual scene analysis, IEEE Transactions On Computers, vol. 20, 1971, 562-569.

APPENDIX III.

Minimal Finite Automata from Finite Training Sets

Department of Mathematical and Computer Sciences
The Atlanta University
Atlanta, GA 30314

Introduction

This paper describes a solution to the following problem: Given a finite set of strings over an alphabet A and a positive integer n , find a deterministic finite automaton (DFA) with a minimal number of states that recognizes the strings in the given set but does not recognize any other strings of length less than n . Clearly such an automaton exists, indeed several minimal automata exist for some combinations of A and n . The chief difficulty in solving this problem was determining a structure that is uniquely determined by the problem data.

This paper begins by expounding standard results on regular sets in a form clearly parallel to the new construction. The construction has been implemented in a version of LISP. Examples mentioned in this paper were derived with this implementation.

Notation

A a finite alphabet

A^* the set of strings over A

ORIGINAL PAGE IS
OF POOR QUALITY

$A(n)$ the set of strings over A of length $\leq n$

$\text{del}(C, a)$... the set of strings s such that as is in C

a is a string over A

Results on Regular Sets

Every regular set is associated with a unique set of regular sets, from which the minimal automaton recognizing that regular set can be easily recovered.

Definition 1: A pointed set of regular sets (PSRS) is a set $\{F_0; F_1, \dots, F_n\}$ of subsets F_i of A^* . F_0 is the distinguished member of the PSRS. A PSRS must satisfy this condition:

For all i , $0 \leq i \leq n$, and all a in A , there is a j , $0 \leq j \leq n$ such that $\text{del}(F_i, a) = F_j$.

Definition 2: A PSRS $\{F_0; F_1, \dots, F_n\}$ is connected if the following condition holds:

For all j , $0 \leq j \leq n$, there is a string s in A^* such that $\text{del}(F_0, s) = F_j$.

Lemma 3: Given a PSRS $\{F_0; F_1, \dots, F_n\}$, let $A = \{a_1, \dots, a_m\}$. Given i, j , $0 \leq i \leq n$, $0 \leq j \leq m$ let $\text{del}(F_i, a_j) = F_{d(i,j)}$. Then $F_1 = a_1 F_{d(0,1)} + \dots + a_m F_{d(0,m)}$.

Proof: O. n. //

Lemma 4: Let $F' = \{F_0; F_1, \dots, F_n\}$ and $G' = \{G_0; G_1, \dots, G_m\}$

be PSRS's. Suppose also that $F_0 = G_0$. Then:

(a) if F' is connected, then F' is a subset of G' .

(b) if F' and G' are both connected then $F' = G'$.

Proof: (a) $F_0 \in G'$ by hypothesis. suppose that for some $p \geq 1$ that the following is true:

(*) For any $s \in A^*$, $\text{length}(s) < p$, $\text{del}(F_0, s) \in G'$.

Let $s \in A^*$ be of length $p-1$ and $a \in A$. Then, by (*), $F_1 = \text{del}(F_0, s) = G_j$ for some i and some j . Then $\text{del}(F_0, sa) = \text{del}(\text{del}(F_0, s), a) = \text{del}(G_j, a) = G_k$ for some k . Thus, (*) is true even if $\text{length}(s) = p$. By induction, $\text{del}(F_0, s) \in G'$ for any $s \in A^*$. Since F' is connected, $F' \subseteq G'$.

(b) Clear from (a). //

Lemma 5: Let $F'' = \{F_0, F_1, \dots, F_r, F_{r+1}, \dots, F_n\}$ be a PSRS. Suppose that for each $1 \leq i \leq r$ there is an $s \in A^*$ such that $F_i = \text{del}(F_0, s)$ and for $r+1 \leq i \leq n$ there is no such s . Then $F'' = \{F_0, F_1, \dots, F_r\}$ is a connected PSRS.

Proof: Suppose that $\text{del}(F_i, a) = F_j$ for some $0 \leq i \leq r$ and $r+1 \leq j \leq n$. Then $F_j = \text{del}(\text{del}(F_0, s), a) = \text{del}(F_0, sa)$ for some $s \in A^*$, contradicting the hypothesis. Thus, F'' is a PSRS. F'' is clearly connected.

Note: A regular set is defined by a regular grammar. We use the definition of a regular grammar as a context free grammar with productions of the form $X \rightarrow aY$ and $X \rightarrow \epsilon$ where X and Y are non-terminals, $a \in A$ and ϵ represents the empty string.

Theorem Let S be a regular subset of A^* . There is a unique

ORIGINAL PAGE IS
OF POOR QUALITY

connected PSRS with S its distinguished member.

Proof: Let S be defined by a grammar over A with non-terminals $\{X_0, \dots, X_n\}$ and start symbol X_0 . For each i , let R_i be the regular set defined by the same grammar, but with start symbol X_i . So, $S = R_0$. Let $F = \{F_0, \dots, F_m\}$ be the set of all unions of the sets R_i , numbered so that $S = R_0 = F_0$. The F is a PSRS. This is seen as follows. For a given X_i and $a \in A$, let $X_i \rightarrow aX_{g(i,j)}$ be all the productions with left hand side X_i and right hand side a followed by a non-terminal. If there are no such productions then $\text{del}(R_i, a)$ is empty, which is one of the unions F_k . Otherwise, $\text{del}(R_i, a) = R_{g(i,1)} \cup \dots \cup R_{g(i,p)}$, which is also one of the F_j 's. Since $\text{del}(C \cup D, a) = \text{del}(C, a) \cup \text{del}(D, a)$, F is seen to be a PSRS. A connected PSRS can be derived by Lemma 5. By Lemma 4, it is unique. //

The Main Problem

Another way of stating the problem we solve here is this. Given a finite set S , of strings over A and a positive integer n , we wish to determine a PSRS $\{F_0; \dots, F_m\}$ with m as small as possible, subject to the condition that $F_0 \cap A(n) = S \cap A(n)$. As mentioned in the introduction, S and n do not uniquely determine such a PSRS. However, the following similar structure is uniquely determined, if one condition be added. From it, all PSRS's that satisfy the conditions can be easily determined.

Definition 7: A pointed set of finite sets (PSFS) is a finite set of

ORIGINAL PAGE IS
OF POOR QUALITY

pairs $\{ (C_0, n_0); (C_1, n_1), \dots, (C_m, n_m) \}$, with distinguished element (C_0, n_0) , satisfying the following conditions:

- (a) n_i is an integer, $C_i \subseteq A(n_i)$.
- (b) For all $0 \leq i \leq m$ and $a \in A$, there is a j such that $\text{del}(C_i, a) = C_j \cap A(n_i - 1)$ and $n_j \geq n_i - 1$.
- (c) For all $0 < j \leq m$, there are i and a such that $\text{del}(C_i, a) = C_j$ and $n_j = n_i - 1$.

Lemma 8: For a PSFS $\{ (C_0, n_0); \dots \}$, $n_i < n_0$ for all $i > 0$.

Proof: Given a pair (C_i, n_i) with $i > 0$, we can build a chain of pairs, by successively applying Definition 7(c), having strictly increasing n_j . As no pair may be repeated in this chain, it must be finite in length. Therefore, it must end with the pair (C_0, n_0) . //

Condition (c) is parallel to the connected condition on PSRS's. However, a further condition is needed to fully minimize a PSRS, given its distinguished pair.

Definition 9: A PSFS $\{ (C_0, n_0); \dots \}$ is minimal if for every i and j , $C_i \cap A(n_j) = C_j \cap A(n_i)$ implies $i = j$.

Theorem 10: Suppose that C' and D' are PSFS's with the same distinguished pair.

- (a) If C' is minimal, then $C' \subseteq D'$.
- (b) If both C' and D' are minimal, then $C' = D'$.

Proof: Let $C' = \{ (C_0, n_0); \dots \}$ and $D' = \{ (D_0, m_0); \dots \}$. Suppose for some n_0 the following is true:

- (*) For every (C_i, n_i) such that $n_{ei} \geq p$, there is a (D_k, m_k) such

that $C_i = A(n_i) \cap D_k$ and $m_k \geq n_i$.

Note that since $C_0 = D_0$ and $n_0 = m_0$ by hypothesis, (*) is true for $p = n_0 - 1$. Let (C_j, n_j) be given with $n_j = p$. Choose (C_i, n_i) and $a \in A$ such that $\text{del}(C_i, a) = C_j$ and $n_j = n_i - 1$. Now, $n_i > p$, so there is a pair (D_k, m_k) as in (*). There is also a (D_1, m_1) such that $\text{del}(D_k, m_k) = D_1 \cap A(m_k - 1)$ and $m_1 \geq m_k - 1$. Then $D_1 \cap A(n_j) = D_1 \cap A(m_k - 1) \cap A(n_j) = \text{del}(D_k, a) \cap A(n_j)$ since $n_j = n_i - 1 \leq m_k - 1$. Thus $D_1 \cap A(n_j) = \text{del}(D_k \cap A(n_j + 1), a) = \text{del}(C_i, a) = C_j$. Further, $n_j \leq m_k - 1 \leq m_1$. So, if (*) is true for some $p < n_0$, it is true for $p-1$. By the remark following (*), it is true for all $p < n_0$. Thus, for any (C_i, n_i) there is a (D_k, m_k) such that $C_i = D_k \cap A(n_i)$ and $m_k \geq n_i$. Since the minimality of C' has not been used, there is, by symmetry, a (D_k, n_j) such that $D_k = C_j \cap A(m_k)$ and $n_j \geq m_k$. Substituting for D_k , $C_i = C_j \cap A(n_i) \cap A(m_k)$ and $n_i \leq m_k \leq n_j$. Thus, $C_i \cap A(n_j) = C_i = C_j \cap A(n_i)$. So, $i = j$ by the minimality of C' . This implies that $n_i = m_k = n_j$ and $D_k = C_i$. That is, every pair (C_i, n_i) is in D' .

(b) From (a), by symmetry. //

As a PSRS can be reduced to a connected PSRS by removing some of its members, so a PSFS can be reduced to a minimal one. However, the removal must be done one pair at a time, as detailed in the following Lemma.

Lemma 1. Let $C' = \{ (C_0, n_0); \dots \}$ be a PSFS. Suppose there are one or more $i \neq j$ such that $C_j \cap A(n_i) = C_i \cap A(n_j)$ and $n_j \leq n_i$. Pick the pair with the smallest n_j . If (C_j, n_j) is removed from C' ,

ORIGINAL PAGE IS
OF POOR QUALITY

the result is a PSFS with the same distinguished pair.

Proof: Call the new set of pairs C'' . By Lemma 8, $j > 0$, so the distinguished pair would not be removed. Note that $C_j = C_i \cap A(n_j)$ and $n_j \geq n_i$. Condition (a) of Definition 7 is clearly satisfied by C'' .

Suppose that, for some k and a , $\text{del}(C_k, a) = C_j \cap A(n_k - 1)$ and $n_j \geq n_k - 1$. Then $\text{del}(C_k, a) = C_j \cap A(n_k - 1) = C_i \cap A(n_j) \cap A(n_k - 1) = C_i \cap A(n_k - 1)$ and $n_i \geq n_k - 1$. Clearly, then, 7(b) is satisfied by C'' .

Suppose that for some k and a , $\text{del}(C_j, a) = C_k$ and $n_k = n_j - 1$. Pick l so that $\text{del}(C_l, a) = C_l \cap A(n_l - 1)$ and $n_l \geq n_i - 1$. Now, $n_l \geq n_i - 1 > n_j - 1 = n_k$, so $C_l \cap A(n_k) = C_l \cap A(n_l - 1) \cap A(n_k) = \text{del}(C_l, a) \cap A(n_k) = \text{del}(C_j, a) = C_k$. However, this contradicts the minimality of n_j , so no such k exists. Condition (c) of Definition 7 is thus satisfied by C'' , it is a PSFS.

//

Corollary 10: Any PSFS contains a (unique) minimal PSFS, with the same distinguished pair, as a subset.

Proof: Repeated applications of Lemma 11 will result in the condition of Lemma 8 being satisfied. Unicity follows from Theorem 10(b). //

We can now describe an algorithm for constructing the minimal PSFS with given distinguished pair (C, n) , $C \subseteq A(n)$. This algorithm maintains two lists of pairs: PROC and UNPROC. Initially, PROC is empty and UNPROC contains only the initial pair (C, n) . The algorithm is this:

while UNPROC is not empty do

 Fetch (D,m) from UNPROC and put it in PROC

 for each $a \in A$ do

 if there is no pair (E, k) in PROC or UNPROC that

 satisfies $\text{del}(D, a) = E \cap A^{(n-1)}$ and

$k \geq n-1$ then

 add the pair ($\text{del}(D, a)$, $m-1$) to UNPROC

Reduce PROC by using Lemma 11.

The final value of PROC, with (C, n) as distinguished pair, is the desired result.

We will show later that the next to the last step (reducing PROC) is unnecessary if UNPROC is treated as a queue. To show that the algorithm actually terminates, we must examine a pair, call it (F, l) = ($\text{del}(D, a)$, $m-1$), added to UNPROC. Assuming $D \subseteq A^{(m)}$, then $F \subseteq A^{(l)}$ since the strings in F are one shorter than those in D. Since UNPROC begins with just (C, n) and $C \subseteq A^{(n)}$, $F \subseteq A^{(1)}$ indeed. Suppose that D were empty. Then $\text{del}(D, a) = D \cap A^{(n-1)}$ and $n \geq n-1$ so no new pair would need to be added. If C itself is empty, then the algorithm only produces one pair, the original (C, n). Otherwise, if F is empty, we may assume that D is not empty and so $m > -1$, that is $l \geq -1$. This shows that the integer part of pairs added to UNPROC are bounded below (by -1 or by n). The integer parts are clearly bounded above by n. Since the strings in any pair are shorter than the longest string in F, there are only finitely many pairs that could be added to UNPROC. The algorithm must eventually terminate.

The algorithm assures that the final set of pairs satisfies 7(b),

by construction. Condition (c) of Definition 7 is satisfied by the manner in which new pairs are introduced.

ORIGINAL PAGE NO
OF BOOK