Complex Preprocessing for Pattern Recognition

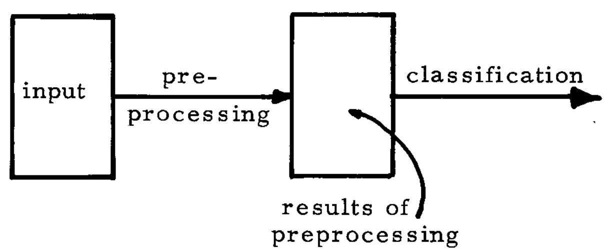
Albert L. Zobrist, University of Southern California, Los Angeles, Calif.

The construction of pattern recognition machines may eventually depend upon the development of highly complex preprocessors. This claim is supported by a discussion of the importance of perceptual grouping. Since complex preprocessing will assess more of the basic structure of a visual scene, internal representations will have to be more descriptive in nature. Two approches to descriptive internal representation are mentioned. Two of the author's programs are reviewed. One plays the Oriental game of GO at a human level and the other can recognize digitized hand printed characters. Both programs use a geometry preserving representation of features, so that calculations involving the features can assess the original geometry of the input. In addition, the GO program calculates groups of stones and performs other types of "complex" processing. Practical and philosophical arguments are given for the use of internal representation by pattern recognition programs.

KEY WORDS AN PHRASES: pattern recognition, perception, preprocessing, perceptual grouping, Gestalt cluster, internal representation, internal model, data structure, character recognizer, machine perception.
CR CATEGORY: 3.63

INTRODUCTION

In the broadest terms, the field of pattern recognition has settled upon a methodology which can be diagrammed as follows:



The preprocessing stage refers to the extraction (or abstraction or calculation) of features (or properties, characteristics, transforms, or measures) which are useful for classification. Preprocessing techniques are often heuristic in nature, reflecting what a program designer feels to be useful. Though the approaches to preprocessing vary widely, they have been collected into a coherent survey by Levine (1). At the other end, classification techniques tend to be more algorithmic in nature. Progress in this area has been surveyed recently by Nagy (2).

Several critiques of the methodology itself appear in the literature. The following two will serve to introduce this report.

Munson (3) likens preprocessing to Art and classification to Science. Since classification seems nearly optimized in both theory and practice, future advances in pattern recognition will come from advances in the art of preprocessing. Two kinds of advances may be expected. First, individual pattern recognition problems will be solved as adequate feature extractors are discovered for those problems. Second, general pattern recognition devices will be built when generally useful feature extractors are discovered. The existence of biological "feature detectors" lends hope to this approach.

Evans (4) states that the model itself is inadequate for some pattern recognition problems, including perhaps the most interesting ones. There appears to be a distinctive gap between the inputs handled by present-day classifiers and complex real world scenes. A structured scene seems to require description, not classification. This requirement was stated early by Greene (5) and Minsky (6).

Instead of proposing the descriptive approach as a second model for pattern recognition, let us consider it to be an excellent direction for current methodology to pursue. This report explores what may be a middle ground between these approaches. "Complex preprocessing" may necessitate descriptive representation of the results of processing. As a start, perceptual grouping is discussed as an example of "complex preprocessing".

PERCEPTUAL GROUPING

Perceptual grouping appears to be the basis of human visual organization. Its importance to human vision has long been recognized by Gestalt psychologists (7). It is responsible for the formation of wholes from parts, hence for the determination of the objects of a visual scene. Objects formed from parts in a scene may themselves be grouped, thus it appears that perceptual grouping yields a hierarchy of parts and subparts. Figure 1 shows some of the physical correlates of grouping. In each case the effect is to give the center column privileged status in the field of vision.

Grouping is not only basic to the structuring of a scene but is usually a prerequisite for the recognition of objects in a scene. The ambiguous figure illusion illustrates this. For example, Figure 2 can appear as a grinning man or as a cherub smoking a cigarette. The figure may be seen one way or the other, possibly in alternation, but not both ways at once. The recognition of man or cherub follows from the determination of figure and ground which follows from the grouping of contours. Figure 3 gives a more pertinent illustration. Both parts of the figure contain a numeral 4, but one is hard to recognize because it is not an object of our perception. The parts of the hidden 4 are distributed among three more primitive objects. The other numeral 4 is easily recognized despite a great deal of visual noise because it is an object of perception.

The determination of relations in the visual field is also due in part to grouping. For example, the leftmost target in a row of targets can be "seen" if and only if the row itself is an object of perception. Experiments with birds (8) have suggested that grouping is essential for this type of relational perception.

One complicating factor is that experience can affect grouping. In some sense, grouping is due to perceptual and cognitive factors, and the two are difficult to separate. Psychologists generally agree, however, that grouping is mainly a perceptual phenomena which helps to organize the visual field. For example, Hebb (9) postulates a "primitive process" of grouping prior to the operation of his perceptual cell assemblies.

If computers are to calculate or describe the structure of visual scenes, then the implementation of perceptual grouping should have high priority. Consider the sequence of grouping problems shown in Table 1. The "perceptual cues" are only rough suggestions as to how grouping might be accomplished.

	Primitive Objects	Groups	Perceptual Cues
1.	points	clusters	proximity
2.	points	gestalt clusters	proximity figure
3.	points	edges lines	proximity good continuation
4.	lines	boxes	corner joins edge matching
5.	lines contours	objects	contiguity wholeness figure
6.	stars	constellations	brightness proximity figure
7.	GO stones	groups armies	color proximity figure

Table 1. Some Perceptual Grouping Problems for Computer Research

The first problem is conceptually the simplest. Namely, the formation of clusters of points in space according to their proximity. This problem has become strongly

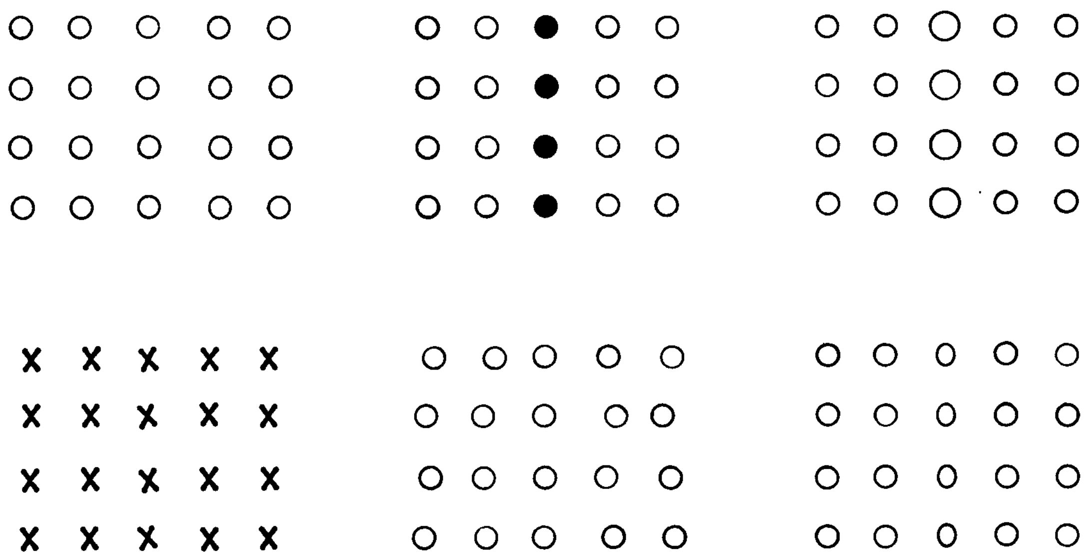


Figure 1. Some Physical Correlates of Perceptual Grouping: Proximity, Size, Regularity, Shape, Color, and Orientation

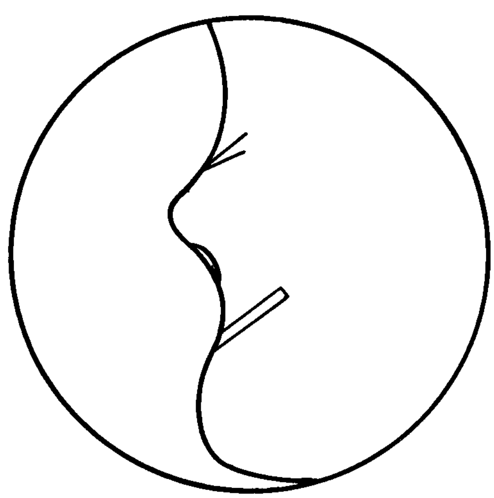


Figure 2. An Ambiguous Figure

associated with the pattern classification problem, where it is assumed that the clusters correspond to pattern classes. Cluster formation without a priori knowledge of which samples yielded which points has been called "learning without reward" by Arkadev and Braverman (10). Zahn (11) has explored some algorithms for the clustering of points in n-space. His basic method is find a type of graph which has the points as nodes, then certain subgraphs correspond to clusters of points. The second problem has also been considered by Zahn, who uses the term Gestalt clusters to mean the clusters which are formed by human vision.

The mobile automaton at Stanford Research Institute (12) groups points into short edges, short edges into lines, and can identify non-overlapping objects bound by those lines. Thus, three levels of grouping are used (problems three and five). Problem three is also of interest to the processing of bubble chamber photographs since the particle tracks are lines or curves formed from bubbles. Problem four was subjected to a nice analysis by Guzman (13). Given a rectilinear line drawing of an overlapping jumble of blocks, his program can sort out the blocks and enumerate their faces. The primary cues are the T and Y joins of the line drawing, where a T join indicates an edge of one block disappearing under another, and a Y join indicates a three-way corner of a single block. Other such corner and edge cues give similar implications. The program then combines all of the implications to decide which faces form which blocks. Good results have been obtained with fairly complicated piles of blocks.

The sixth problem has not been attacked. However, it might provide an interesting test for any computer algorithms which are dedeloped — especially if the algorithm produced the same groups of stars without a priori knowledge of the figures (dippers, lions, etc.) they represent to us.

The role of perceptual grouping for the Oriental game of GO has been discussed by the author in an article (14) which also describes the rules of GO and a program which plays GO at a human level. It is sufficient to say here that a GO game is seen as a 19×19 array of black and white stones, and that recognition of the groups of stones of the same color is very important to good play.

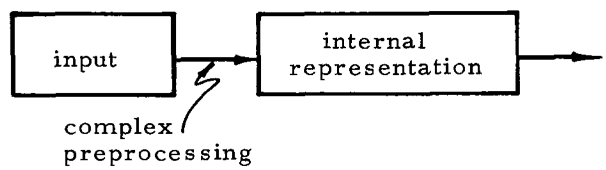
The GO program uses positive and negative numbers for the two colors of stones, and a "smearing" of the numbers to calculate the groups. Figure 4 shows the smearing operation for a corner of a GO board. The method is suggestive of potential functions or heat flow.

The GO program and the robot projects have one common aspect. They not only apply complex processing to an input to produce a description, but the description is used for difficult problem solving tasks.

The programs just described are relatively complex, and it is likely that they will become more so. Thus it appears that future pattern recognition programs (or machine perception devices) will require what has been called "complex preprocessing" in the Introduction. Nor is the implementation of perceptual grouping the only source of complexity. Other perceptual abilities, such as the ones which make a master chess player, may be very complex indeed.

INTERNAL REPRESENTATION

This section considers some general implications of complex preprocessing for the pattern recognition methodology outlined earlier, which can now be diagrammed as follows:



Serious thought must be given to the nature (data structure) of the internal representation. The most cited approach is to transform an input into a vector in n-space using n feature detectors, each of which computes a number. This approach has been satisfactory for simple feature extractors and for the relatively simple problem of pattern classification. It has also satisfied the gut feeling that data compression (or abstraction) must take place. That is, a point in n-space is more compact than a visual scene. The development of machine perception may hinge upon the development of more adequate internal representations. The diagram above suggests two important considerations. First, the data structure must be capable of representing the results of complex processing. Second, it must be convenient for further processing. To rephrase these considerations, a rich representation is necessary to convey the richness of perception, and the nature of the representation determines the amount and quality of further processing. In view of these considerations, one should no longer expect that an internal representation be more compact than a "raw" pattern itself.

Two approaches to this problem can now be seen in the literature. The first is to develop some sort of language which is capable of describing the structure of a picture.

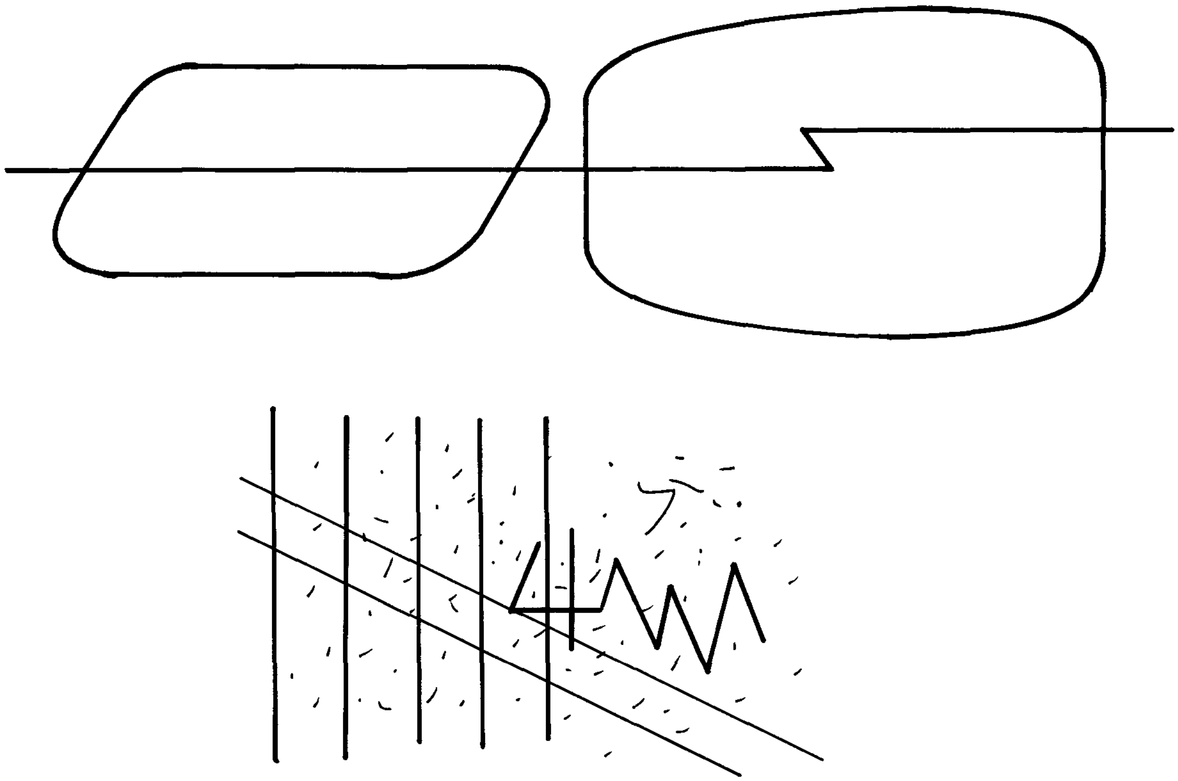
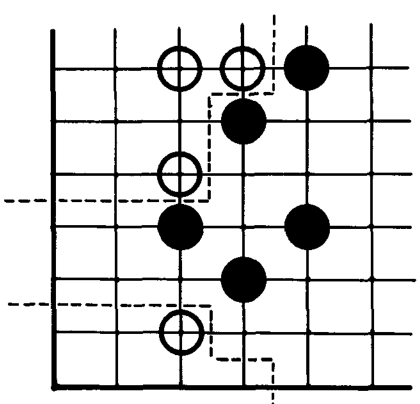


Figure 3. Illustration of the Role of Perceptual Grouping in Recognition



(a)

.	.	-50	-50	50	.
.	.	.	50	.	.
.	.	-50	.	.	.
.	.	50	.	50	.
.	.	.	50	.	.
.	.	-50	.	.	.
.	.	.	.	.	.

(b)

.	-1	-51	-49	49	1
.	.	-1	49	2	.
.	-1	-49	.	1	.
.	1	49	3	50	1
.	.	1	50	2	.
.	-1	-50	.	.	.
.	.	-1	.	.	.

(c)

-1	-2	-55	-49	51	2
.	-3	-2	48	4	2
-1	-1	-50	2	3	2
1	1	51	6	54	2
.	1	1	53	4	2
-1	-2	-51	.	1	.
.	-2	-2	-1	.	.

(d)

-2	-6	-59	-49	53	4
-3	-6	-4	48	8	6
-1	-3	-50	4	7	5
1	3	53	10	58	6
1	2	4	56	8	4
-2	-4	-52	.	2	2
-2	-4	-5	-2	.	.

(e)

-5	-10	-63	-49	55	8
-6	-10	-6	48	12	10
-2	-5	-50	6	11	9
2	5	55	14	62	10
2	4	6	59	12	8
-3	-6	-53	.	4	4
-4	-7	-8	-3	1	1

(f)

Figure 4. Illustration of the Smearing Method Used for Grouping by the GO Program.

An early pattern recognition program by Grimsdale (15) described hand printed characters in terms of their strokes. The Geometry Analogy of Evans (16) relied heavily on descriptions of the structure of inputs. Following Narasimhan (17) and Kirsch (18), many workers have investigated the use of formal languages for description and parsing methods for the generation of descriptions.

Though many of these programs give a description of an input as the final output, the description can be regarded as a representation of the input for further use. Formal descriptions suggest theorem proving as the basis of further processing. Thus, theorem proving robots may develop from this approach.

The second approach is to give the representation a structure which reflects the important structure of the input. This approach could be used to represent structure which is hard to translate to other forms or if translated, is hard to manipulate. A candidate for this approach is geometric structure.

The author's GO program (16) uses a geometric representation of features. That is, the 19 x 19 GO board is translated to twelve 19 x 19 arrays of features. (Figure 4 shows the calculation of only one of the twelve arrays.) If a certain feature occurs in the lower left corner of some array of the internal representation, then a mark appears in the lower left corner of an array of the internal representation. If it is desired to check for the adjacency of two features, then it is sufficient to check for their marks, making sure that they correspond to adjacent locations. This latter task is handled by templates. The templates are able to assess structure of the input by referencing the internal representation. For a specific example, refer to Figure 4. A simple template might search for a positive number adjacent to a negative number. The matching of this template indicates a border between opposing groups of stones. Given that the array in Figure 4 represents groups, then "border" is a higher level feature which happens also to be conveniently represented. If the geometric structure is translated into a list structure, then the discovery of borders might require list searching. If the geometric structure is expressed in some formal language, then the equivalent of theorem proving might be required.

The preservation of geometric structure through layers of processing is in general agreement with physiological studies of the visual system of mammals. For example, Hubel and Wiesel (20)have traced the response to visual stimuli as far as the occipital region of the brains of cats. They have found impulses in the striate cortex which correspond to features in the visual field such as spots or edges, and that the mapping from retina to cortex is roughly geometry preserving. The existence of a biological prototype lends support to the second approach.

A CHARACTER RECOGNITION PROGRAM

To test the feasibility of geometry preserving internal representation for character recognition, a simple program was written in FORTRAN-V for the UNIVAC 1108 computer. The preprocessing part of the program was borrowed from Uhr-Vossler (21). The program's operation can be outlined as follows:

Step I. Uhr-Vossler operators are applied to the input. Whenever they match, a mark is placed in the corresponding point of an array reserved for that operator.

Step II. Templates are scanned over the arrays produced by Step I. Each template requires its marks to be found in their arrays at fixed relative position. The templates imply the names of the alphabet with certain weights.

Step III. The weights are summed. The name with maximum weight is chosen as a response.

Step IV. The program is given the correct response as feedback. The values and the weights of implication of the templates are adjusted up or down, and the values of the Uhr-Vossler operators are also adjusted.

Step V. If the feedback is different from the response, then new Uhr-Vossler operators and templates are extracted from the input. Operators and templates with low value may be discarded.

Steps I and II are illustrated by Figure 5. The Uhr-Vossler operators are 5 x 5 arrays of zeros, ones, and blanks. Each operator is scanned over the input, and when the zeros and ones all match then a bit is placed. The placement is at the row and column corresponding to the center of the 5 x 5 array during a hit. For the results shown in this section, between 64 and 84 operators were used. The 64 to 84 arrays produced by their application constitute an internal representation of the input.

A template consists of n references to the internal representation together with a specification of the geometric arrangement of the n-tuple. The 2-tuple shown in Figure 5 can be expressed as follows:

(0, 0)	operator 1
(3, -1)	operator 2

Each template is scanned over the proper arrays to find a match, but no rotations or reflections are used. The geometric relations are allowed to wobble a distance of 1. For example, the bit placed by operator 2 could be at relative location (4, -2) in Figure 5.

Each template has an associated weight vector containing one weight of implication for each letter of the alphabet. If all parts of a template match, then its weight vector is added to a total sum vector. The highest number in the total sum vector determines the response of the program. The program can then create, adjust, and discard operators or templates using methods similar to those adopted by Uhr and Vossler. Precise details are given in another report (22). The borrowing of techniques from a well known program was done to focus attention on a major difference between the two programs. Considering the 5 x 5 windows to be feature detectors, the programs use different representations of the results of feature detection. The Uhr-Vossler program produces a list of numbers which give information on the location of features. But since the information is in a simple list, it is relatively difficult to assess the geometric relations (above, to the left of, etc.) between features. The geometric internal representation used by the new program allows the templates to assess this structure in a natural fashion. (For those who are unconvinced, consider the border detector mentioned above in connection with the GO program. If the positive and negative portions of the board are stored in lists then the border detector would be approximately 120 times slower.)

The program was tested on the Highleyman data (23) which consists of 50 alphabets of hand printed characters, each alphabet having 10 numerals and 26 upper case letters. Each sample character is quantized and encoded as a 12 x 12 binary array. Table 2 compares the program's performance with that of other programs (24, 25, 26, 27) and with human performance (28). The 34 letter alphabet omits the numerals 0 and 1. A 10 letter alphabet refers to the numerals only. In most experiments, the first forty alphabets were used for training and the last ten alphabets were used for an independent test.

Two performance measures should be noted. First, the recognition rate on the independent test indicates whether the program can generalize from training data to test data (i.e. pattern recognition = stimulus generalization). Second, the generalization decrement indicates how efficiently it generalized. In both of these respects the Zobrist program performs very well. The performance is especially notable since the program is a learning program of the Uhr-Vossler type (feature discovery). Thus, the

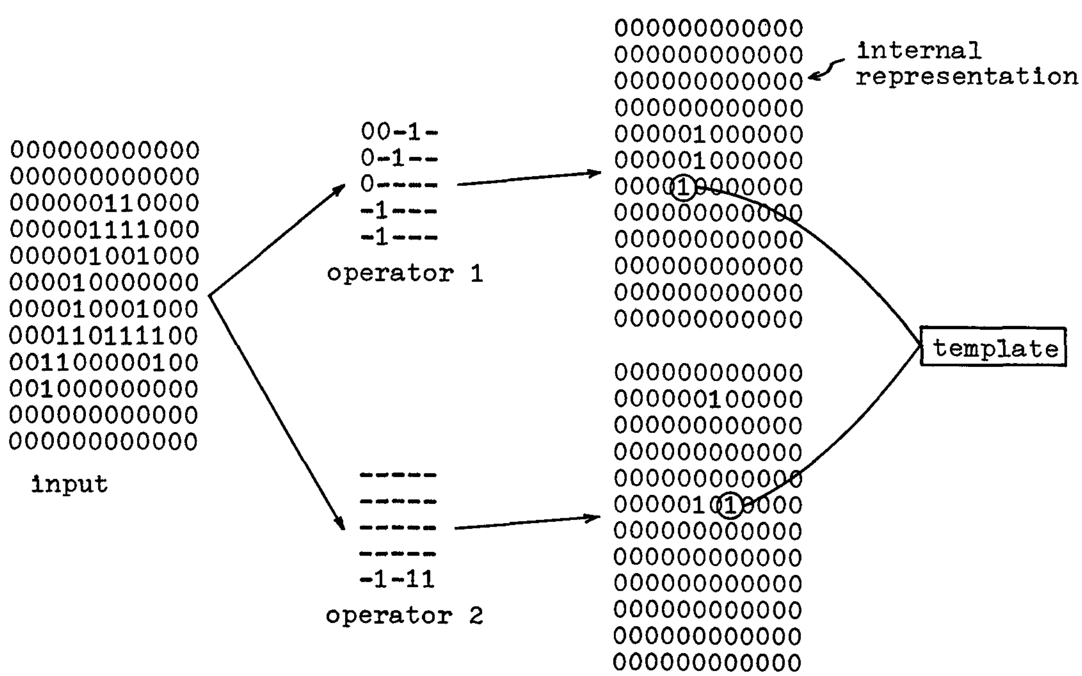


Figure 5. Illustration of the Scanning of Uhr-Vossler Operators and Templates by the Author's Program

		Year	Size of Alphabet	Training Score	Test Score	Generalization Decrement
1.	Zobrist	1969	34	49.4	43.8	5.6
2.	Zobrist	1969	10	76.8	63.0	13.8
3.	Zobrist	1969	10	83.3	75.0	8.3
4.	Munson	1968	36	-	68.3	-
5.	Munson	1968	10	-	88.0	-
6.	Chow	1962	36	93.3	58.3	35.0
7.	Bledsoe	1961	36	71.0	25.0	46.0
8.	Bledsoe	1961	36	87.0	29.0	58.0
9.	Bledsoe	1961	36	98.0	31.0	67.0
10.	Bledsoe	1961	36	80.0	32.0	48.0
11.	Bledsoe	1961	36	86.0	40.0	46.0
12.	Highleyman	1961	36	77.2	-	-
13.	Highleyman	1961	10	83.0	-	-
14.	Nearest Neighbor	1968	36	100.0	52.5	47.5
15.	Human Performance	1968	36	-	84.3	-
16.	Human Performance	1968	34	-	88.5	-
17.	Human Group	1968	36	-	88.6	-

Table 2. Recognition Rates for Experiments with the Highleyman Data.

geometry preserving representation used by the GO program seems to work quite well for character recognition.

PRACTICAL AND PHILOSOPHICAL ARGUMENTS FOR INTERNAL REPRESENTATION

The role of internal representation in pattern recognition methodology has been relatively ignored. For example, in the area of problem solving, internal representation is considered all-important and a distinction is made between internal and external representation (Ernst and Newell, (29)). In fact, pattern recognition exhibits a slight tendency in the other direction. Feigenbaum (30, p. 1011) paraphrases one group of robot builders as saying, "The most economic and efficient store of information about the real world is the real world itself." Arguments against the use of internal representation seem to be based upon expediency. If a computer is fitted with a video input, then the easiest thing to do is use the real world as bulk storage. This section presents some arguments in favor of the use of internal representation.

The first involves the issue of "calculation vs. storage". Given that the preprocessing stage of pattern recognition becomes expensive in time or effort, then it should be more efficient to store the results rather than run the risk of having to recalculate. This should be especially true of preprocessing which corresponds to the basic visual organization of a scene. A good example is the grouping performed by the GO program described earlier, which takes about .1 second of CPU time to perform. The program refers to this information hundreds of times in the course of calculating where to move. Thus, storage of the information on groups saves a great deal of computer time.

The second argument is that internal representations or models are fundamental to intelligence. This fits nicely with Craik's (31) hypothesis on the nature of thought, which can be paraphrased as follows. One of the most fundamental properties of thought is its power to predict events. The thinker accomplishes this by means of three fundamental processes:

1. "Translation" of the external world into internal symbols,
2. Arrival at other symbols by calculation or symbol manipulation, and
3. "Retranslation" of the resultant symbols to the external world.

These processes are diagrammed in Figure 6. The dashed arrow corresponds to external events which are isomorphic to the internal calculations performed by Step 2. These could be potential events predicted by the thinker or real events recognized by the thinker. According to this view of intelligence, the amount and quality of processing possible in Step 2 depends upon the nature of the representation created by Step 1. Craik's hypothesis not only stresses the importance of internal representation, but suggests its role in a unified approach to

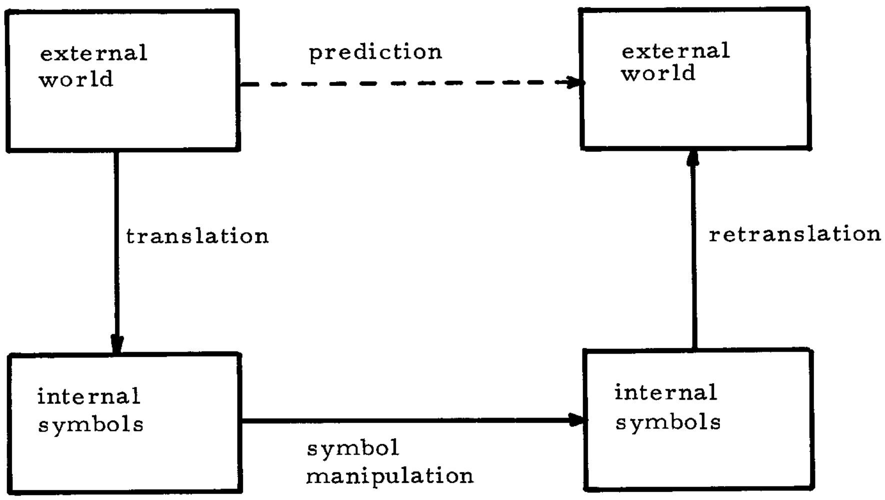


Figure 6. Diagram of the Three Processes which Enable an Intelligent Being to Predict Events in the External World

artificial intelligence. Pattern recognition work should be concerned with the representation of features or properties for subsequent processing; problem solving work should be concerned with the use of representations which can be extracted from the real world.

BIBLIOGRAPHY

1. Levine, M. D., "Feature extraction: a survey", Proc. IEEE, vol. 57, no. 8, pp. 1391-1407, August 1968.

2. Nagy, G., "State of the art in pattern recognition", Proc. IEEE, vol. 56, no. 5, May 1968.

3. Munson, J. H., "Some views on pattern recognition methodology", Proc. Internat'l Conf. on Methodologies of Pattern Recognition, Univ. Hawaii, January 1968.

4. Evans, T., "Descriptive pattern analysis techniques: potentialities and problems", Proc. Internat'l Conf. on Methodologies of Pattern Recognition, Univ. Hawaii, January 1968.

5. Greene, P. H., "An approach to computers that perceive, learn, and reason", Proc. WJCC, 1959.

6. Minsky, M., "Steps towards artificial intelligence", Proc. IRE, vol. 49, no. 1, January 1961.

7. Kohler, W., Gestalt Psychology, 1947, Liveright, New York.

8. Hertz, M., Zeitsch. f. vergl. Physiol., 1928.

9. Hebb, D. O., Organization of Behavior, 1949, Wiley, New York.

10. Arkadev, A. G. and Braverman, W. M., Computers and Pattern Recognition, 1967, Thompson, Washington, D. C.

11. Zahn, C. T., "Graph-theoretical methods for detecting and describing gestalt clusters", IEEE Transactions on Computers, vol. C-20, no. 1, January 1971.

12. Nilsson, N. J., The Mobile Automaton: An Application of Artificial Intelligence Techniques, Stanford Research Institute, Menlo Park, California, March 1969.

13. Guzman, A., Some Aspects of Pattern Recognition by Computer, Ph. D. Thesis, 1969, MIT, Cambridge, Massachusetts.

14. Zobrist, A. L., "A model of visual organization for the game of GO", Proc. AFIPS SJCC, vol. 34, pp. 103-112, 1969.

15. Grimsdale, R., Sumner, F., Tunis, C. and Kilburn, T., "A system for the automatic recognition of patterns", Proc. IEEE, vol. 106B, no. 26, pp. 210-221, March 1959.

16. Evans, T., A Program for the Solution of a Class of Geometry-Analogy Problems, Ph. D. Thesis, Dept. of Mathematics, 1963, MIT, Cambridge, Massachusetts.

17. Narasimhan, R., "Syntax-directed interpreted of classes of pictures", Comm. ACM, vol. 9, no. 3, March 1966.

18. Kirsch, R., "Computer interpretation of English text and picture patterns", IEEE Trans. on Elec. Computers, vol. EC-13, no. 4, August 1964.

19. Raphael, B., "Programming a robot", Proc. IFIP Congress 68, August 1968, Edinburgh, Scotland.

20. Hubel, D. H. and Wiesel, T. N., "Receptive fields of single neurons in the cat's striate cortex", J. Physiol., vol. 148.

21. Uhr, L. M. and Vossler, C., "A pattern recognition program that generates, evaluates, and adjusts its own operators", Proc. WJCC, 1961.

22. Zobrist, A. L., "The organization of extracted features for pattern recognition", Journal of the Pattern Recognition Society, (in press).

23. Highleyman, W. H., "Data for character recognition studies", IRE Trans. Electr. Computers, vol. EC-12, p. 135, April 1963.

24. Munson, J. H., Duda, R. O., Hart, P. E., "Experiments with Highleyman's data", IEEE Trans. Electr. Computers, vol. EC-17, pp. 399-401, April 1968.

25. Chow, C. K., "A recognition method using neighbor dependence", IRE Trans. Electr. Computers, vol. EC-11, pp. 683-690, October 1962.

26. Bledsoe, W. W., "Further results on the n-tuple method", IRE Trans. Electr. Computers, vol. EC-10, p. 96, March 1961.

27. Highleyman, "An analog method for character recognition", IRE Trans. Electr. Computers, vol. EC-10, p. 96, March 1961.

28. Munson, J. H., "Experiments in the recognition of hand-printed text: part I", Proc. AFIPS FJCC, pp. 1125-1138, 1968.

29. Ernst, G., Newell, A., GPS: a case study in generality and problem solving, ACM Monograph Series, 1969, Academic Press, New York.

30. Feigenbaum, E. A., "Artificial intelligence: themes in the second decade", Proc. IFIPS 68, North-Holland, Amsterdam, 1969.

31. Craik, K. J. W., "Hypothesis on the nature of thought", The Nature of Explanation, 1943, Cambridge Univ. Press, Cambridge.