Artificial Patterns

ALLEN KLINGER, MEMBER, IEEE

Abstract—This paper deals with obtaining a data base for test and 
evaluation of large software systems, when at most, only a few typical 
pattern instances are available. Data structure variation is proposed 
and demonstrated for some simple images. Relationships to funda-
mental concepts in computer science methodology are discussed.

Index Terms—Data base, data structure, heuristic search, image 
processing, modeling, noise, pattern recognition, simulation, speech 
recognition, statistics.

I. INTRODUCTION

A "PATTERN" is an element of a set of objects which in 
some useful way can be treated alike [1], [2], such as 
all alphabetic letters "A," several photographs of a face, 
normal electrocardiograms, and business-indicator values 
during stock market booms. In practical situations, single 
instances or relatively small samples of a pattern are usually 
available, yet statistical pattern classification theory assumes 
large numbers of cases. Large costs can be incurred for collec-
tion of more data—and in some cases, it may be impossible 
to obtain other sample patterns.

This paper calls real data sources "natural patterns" and 
focuses on computer generation of artificial patterns. The 
purpose is to obtain many varied, yet representative patterns 
for testing recognition programs. In this attempt to address 
the real need for software testing, evaluation, and verifica-
tion tools, we examine some fundamental pattern recognition 
concepts, and relate these concepts to other computer science 
methodology areas.

Digital computer methods to produce artificial patterns 
involve simulating or modifying natural patterns, and this 
can be done only after analysis. Since the concept of 
"pattern" is operational, two basic research concepts, the 
notion of "feature" and the distinction between global and 
local decision-making, must be used in the analysis process. 
The illustrations and examples touch on how these are inter-
related in some simple image patterns.

Although this paper describes prototype software to vary 
image patterns, there are relationships to management informa-
tion systems through "key" data and arrays, and limited 
company operating modes. Images are digitized to arrays 
and visual "features" are like keys. Data management systems 

and pattern recognition programs both involve multiply-
connected multidimensional data. Often, only a few cases 
such as values of warehouse inventories, suppliers' productions, 
and demands, all at monthly intervals, are on hand for test 
and evaluation of software. Obtaining a larger set of test 
data is very desirable to see that the software can withstand 
demand-peaking or rapid growth situations: realistic evalua-
tion of management information systems is behind the 
following efforts to vary image patterns.

Software engineering for generation of artificial patterns 
expands procedures used to create the statistical simulation 
tool [3], a large random-number table. The concern here is 
with adding randomness to patterned data within structure, 
or creating deliberate distortions of either features or local 
and global relationships, but keeping pattern type unchanged. 
In [3], statistical tests were applied to numbers from phys-
ically random events. Here, variation is made of physically 
patterned data to create larger data bases of patterns. Methods 
to randomize or distort patterns are based on data structures. 
Sections on data bases, data structures, and algorithms and 
experiments follow.

II. DATA BASES AND DATA STRUCTURES

Creation of a data base (set of records, frequently of similar 
items) to test classification procedures is of general concern. 
Nine involving natural patterns ranging from alphanumeric 
characters (printed and handwritten) to speech data are dis-
tributed by the IEEE Computer Society [4]. The cost of 
compiling a data base can be high and the task onerous. 
To obtain a large number of sample speech patterns, many 
speakers must spend time in front of a recording microph
ne, the resulting audio tape must be converted into a digital 
tape, and specific patterns must be distinguished and labeled 
by content and speaker.

By contrast, artificial creation of a data base requires cre-
ating a "pattern data structure"—a description of the key 
elements of the pattern and their interrelationships in the 
computer. Although the actual representation may depend 
on the machine, the form: linear list, tree, ring, etc., [5] 
depends mainly on the pattern, its permissible variants, and 
other patterns to which it may be compared. A data structure 
[5] is a preferred pattern representation since it enables 
many kinds of computations. A mixture of statistical [2] 
and structural [6] pattern recognition methods is needed to 
solve practical problems [7] where data structures facilitate 
class-recognition computations, or intermediate steps: feature 
and primitive evaluation. Tests of many hypotheses [8] and 
repeated processing of patterned data [9]–[11] are charac-
teristic in realistic nonpictorial cases. Since an image yields
a very large digital record, methods that locate lines and regions [12], [13] are used first in pictorial pattern recognition.

Artificial patterns to expand a small data set into a data base can be based on pattern structure. Thus, a simple deformation such as changing line presence or position: "E" becomes more like "F" or "K" more like "X", can be used. Further discussion of patterns' data structure appears in [14]-[16], artificial generation of deformed patterns is discussed in [17], [18], while [19] uses pattern data structure for picture information reduction. More traditional methods for reducing a pictorial record create an artificial pattern called a "line drawing," usually by gradient-like means [12], though other "edge operators" [20] are also used. The essence of the method is combining a picture element (pixel) with its immediate eight neighbors (see Fig. 1); the result is a value in the derived pattern.

Many types of transformations are used to derive secondary images from natural patterns to enhance or locate structure. Two-dimensional Fourier and Walsh-Hadamard transforms are best known, while in speech, the univariate versions are used along with the logarithm. These transformations yield artificial patterns whose main purpose is as data structures. Thresholding is also used to discard randomness or enhance structure. Fig. 2 shows the marked structural differences present in Fourier transforms of alphabetic characters, while Fig. 3 gives the effect of thresholding to aid in detecting a sphere from a noise background. For other artificial patterns from realistic pictures and speech, see [21, Figs. 21-23, 33(c)-(f), and 41]. In all these cases, the artificial patterns are used as data structures for evaluating features, and data bases are created from records in the derived form.

### III. Algorithms and Experiments

Methods to generate artificial patterns by deforming an ideal source require a pseudorandom number generator and specification of how noise is to be added. In Fig. 4, a more realistic image of a tank is shown that was obtained by adding digits

![Fig. 1. Eight neighbors of a cell.](image1)

![Fig. 2. The letters F, E, C, and Q and their spatial Fourier transforms.](image2)

![Fig. 3. "Noisy sphere": patterns derived from a probabilistic disk (A. Rosenfeld).](image3)

the input for edge lines "stripes." Here the natural patterns are straight-line alphanumeric characters digitized on 12 X 12 arrays, and the heuristic locates points to be randomized usually when three out of six of any straight-line sequence of pixels is found "set" (some experiments used threshold two).

---

1 See Acknowledgment section.
Fig. 4. Noisy tank artificial pattern.

Fig. 5. Random deletion from solid letters.

Fig. 6. Random insertion and deletion.
In these figures, the ideal start pattern had adjacent pixels set, so the heuristic was well matched to the search problem. The same effect can be obtained for more complex cases by replacing the six straight-line adjacent pixels by sets that could appear in edges due to quantization effects (see Fig. 7). Some simple variants: basic, continuous, and discontinuous groups of six pixels are presented in Fig. 7(b). The generic term "stripe" is used there and below to denote a set of pixels likely to occur in an edge of a natural pattern, and hence, used as a search-set in a heuristic procedure to detect structure. The algorithm that uses this idea follows.

**Algorithm:** Random variation in detected structures. This procedure searches an array for stripes, relatively contiguous pixel-sets, with more than a certain fraction dark. A pseudorandom number generator determines which dark pixels are retained. Probability of retention is an input control parameter, as is stripe geometry. Different fraction thresholds can be used for different stripes. A second pass pseudorandomly inserts dark pixels in blank stripe elements, provided the stripe originally exceeded the fraction threshold. Insertion probability is an input parameter; if it is not set, a default condition $p(\text{insert}) = 1 - p(\text{delete})$ is used.

An artificial data base can be generated from a small set of natural patterns by bootstrapping using this algorithm. Since any of the generated patterns in Figs. 5 and 6 could also be used as inputs to the algorithm, repeated cycling of output back to input can cause rapid growth in the number of distinct pattern versions. Thus, a large artificial data base can be developed from a small set of natural patterns. This is similar to the use of a "seed" in modern pseudorandom number generation algorithms used for statistical simulations.

Note that local properties other than solidity are frequently present in similar natural patterns. Fig. 8 shows four sets of $F, E, C, O$ letters with different gross structures. The last set seems blurred by comparison with the first, yet people easily recognized all. Similarly, people perceive boundaries in coastlines and terrain cover changes (as at timberline) in spite of irregularities, and pseudorandom or deterministic change of local detail can be used to generate artificial patterns. Related types of distortion are blurring, where pixels
in or around the original prototype are changed, and stretching. Fig. 9 shows these and contrasts them with array noise, (see Fig. 3 for an example). Note that array noise does not depend on the pattern. Rotation, relocation, and noise only within the pattern are other possibilities. These last three cases and the blurred waveform of Fig. 10 appear in [18].

IV. Conclusions

Artificial patterns have been generated from natural patterns of similar type by prototype software. Heuristic methods were used to detect structures in patterns, and simulation used to create more realistic images from ideal inputs.

The experiments bear out the conclusion that a small set of natural patterns can be used to build a large data base of similar artificial patterns. Several traditional procedures for rearranging natural patterns, including Fourier transformation, were described and the data structure aspect of these artificial patterns was discussed.

Since causality, the relation of observed patterns to physical processes is missing in most natural patterns, algorithms to create artificial patterns could provide models and theories. Most domains where pattern recognition is applied do not possess an accepted theoretical foundation. In neural modeling [22], equations by Hodgkin and Huxley [23] became the standard reference for subsequent researchers. The equations began as a model and became the best available theory. In a similar way, an artificial pattern, or the data structure and algorithm used to assemble it, may be the key to new knowledge.

ACKNOWLEDGMENT

The Fourier transforms of alphabetic letters were laser-produced at the Naval Electronics Laboratory, and communicated by J. Vidal; the noisy-sphere pictures were communicated by A. Rosenfeld. W. Adkinson programmed and ran a version of the algorithm that generated the noisy alphabetic characters; R. Hooper developed the blurring and rotating algorithm; V. To obtained the noisy tank. Their contributions are gratefully acknowledged.

The author expresses his appreciation for this support to the agency and the project monitors, Col. T. Wachowski and Col. G. McKemie, who both made constructive remarks as this work progressed.

REFERENCES


[22] *Proc. IEEE*, vol. 56, June 1968 (Special Issue on Studies of Neural Elements and Systems).


Allen Klinger (S’56–A’59–M’66), for a photograph and biography, see p. 161 of the March 1977 issue of this *Transactions*.