

PATTERN: LINGUISTIC ANALYSIS INVARIANT FOR PLANE TRANSFORMATIONS

J. C. Simon and A. Checroun

Institut de Programmation - 9, quai Saint Bernard
PARIS 5e

Pattern analysis using syntactic methods is discussed briefly to introduce the ideas of the linguistic nature of the attributes and of possible invariance properties versus usual geometrical transformations.

A linguistic operator invariant for projections from plane to plane is proposed and applied to the analysis of point patterns X . From knowledge of the operator "result" or "attribute", it is possible to obtain the convex envelope, reconstitute X , compare X to another pattern X' and more generally, to obtain the possible common subpatterns. This last process shows that a threshold exists, under which a pattern comparison is not reliable.

Linguistic operators invariant for plane similarities are also considered.

§1. INTRODUCTION

1.1. FORMALIZATION OF PATTERN RECOGNITION OPERATIONS

Except for some very simple recognitions, many "recognition levels" have to be distinguished in a pattern recognition problem.

Recognition operations have to be performed to get from one level to the next. Most of the time, they may be described under a common formalization:

- Let a pattern X be a set of "primitive" patterns, each of which has a name to which are associated numerical values.
- One or more analysis operators $\&$ operate on X through algorithms; the result of $\&$ is u_X .
- A recognition decision is performed by comparing u_X with information previously stored. If this phase is successful, a pattern "feature" is extracted. This feature has a name; numerical values may be associated with it.

The recognized features are the new primitive patterns. Their set is the pattern on which the operators of the next level will operate.

This scheme is valid obviously for the first level extraction of simple features where X is the set of samples obtained through a measuring instrument such as an optical or acoustical device. Most of the time, the operators $\&$ are linear operators. For example, let X be a one dimensional "signal", i is the name of the primitive measure or sample, x_i the numerical value associated with it.

Let $\&$ be the operator such that the result u_X^p is given by $u_X^p = \sum_{j=1}^{j=q} \alpha_j x_{p-j}$ with $q \ll n$, $1 \leq p \leq n - q$.

The operators $\&$ are defined by the set of numerical values $(\alpha_1, \alpha_2, \dots, \alpha_q)$. The result u_X^p is the numerical value obtained by a translation of p units and a scalar multiplication. Digital filters, neurone nets and Perceptrons make use of such linear operators.

The recognition decision is taken according to the numerical value of u_X^p . Usually if u_X^p is greater than a threshold u_0 , the feature corresponding to $\&$ is recognized.

Such a process is easily extended to images, i.e. two dimensional signals.

1.2. SYNTACTIC OR LINGUISTIC OPERATORS

It has been realized for quite a time that linear operators alone are not able to take into account structural relationships in patterns.

At the beginning of the 1960's a number of authors such as Eden (1), (2), Minsky (5), Narasimhan (6), Kirsch (4), advocated a "description-processing" of pattern, thus starting a new "syntactic" approach to pattern recognition problems of higher complexity. Quoting FU and SWAIN (3), who recently reviewed the state of the art:

"Syntactic pattern recognition is an attempt to adapt the techniques of formal language theory which provides both a notation (grammar) and an analysis mechanism (parsing) to the problem of representing and analysing patterns... Syntactic analysis can proceed only if a grammatical model for the data generation and / or analysis process can be formulated (This does not necessarily imply that the data is actually generated by a mechanism which operates in the same way as the model). A related practical problem is the development of efficient analysis procedures based on the grammatical model. The available literature concerning syntactic pattern recognition deals almost exclusively with the formulation of grammatical models, although the development of analysis algorithms is currently receiving more attention."

This new approach is a subject of considerable interest among pattern recognition specialists. Nevertheless, the practical results in the analysis field do not seem up to the expectations.

In fact syntactic techniques are successful, says Kirsch, only if the generative grammar is fully known. But, as for natural languages, there are some doubts about the existence of such "CHOMSKY-like" grammars for images, even with the improvements provided by the "web grammars" proposed by Pfaltz and Rosenfeld (7).

We would like to retain two features of the syntactic approach:

- Give up the interpretation of the processed quantities as numerical values, consider them rather as words of a language with which numerical values may be associated.
- Try to find "rules" valid for a large class of patterns.

Accordingly, coming back to our formalism, we propose to consider analysis operators $\&$ of a linguistic and invariant nature:

- The result u_X is to be one or more words built on the alphabet consisting of the names of the primitive patterns. For this purpose a formal generative grammar is not absolutely necessary, as long as an algorithm giving an appropriate result can be found. This algorithm, called the analysis operator, should be implemented, of course, by a computer program. The pattern X will be the variable data set of this program.
- The result u_X is to be invariant under some transformations of the set X , corresponding to some geometrical transformations of the analysed object. This "invariance concept" underlies the syntactic pattern methods, though previous authors do not seem to emphasize its relevance.

The recognition decision will be taken by comparing the u_X with some stored information (I.1. phase c). We will see later examples of how this phase can be implemented.

I.3. REMARKS

1. Our formalism can be compared with the one proposed by Fu and Swain (7). For them a Generalized Syntactic Formalism (GSF) utilizes :

- A Generalized Syntactic Element (GSE), which is a construct consisting of two parts : a name and an attribute list, which is a set of modifiers or variable properties.
- A Generalized Syntactic Production (GSP), which is a rule specifying how a syntactic element is composed from or generates other syntactic elements. Usually the syntactic elements generated are "more primitives".

Thus, a set X of primitive patterns can be compared to a GSE, an analysis operator & to a GSP with the difference that now the results are "less primitive" elements, "result" and "attribute" being equivalent words.

Our terminology seems more general, being also appropriate for the filtering or sifting processes. One could in fact assert that in terms of the computer states there is no basic difference, only the semantic differs.

To stress the fact that no generative grammar is implied by our operators, we would prefer to call them "linguistic" rather than syntactic.

2. An instrument provides numerical results or samples ; a name is given to each sample and a number of numerical values. For example, in a retina the cell of name i gives an intensity of numerical value z_i . The coordinates of cell i are the numerical values x_i, y_i . Thus an "elementary measure" is defined by i, x_i, y_i, z_i . The set of these measures is the first pattern, each "elementary measure" being a "primitive pattern". This level will be called the first level. It is already a "feature level", if we consider that the "elementary measures" are the "results" of some "physical

operators". But usually we cannot modify the qualities of these physical operators. This property could be used as a way to define the "outer world" or "universe"...

The second level is reached, most of the time by the use of linear operators or filters. These simple operators have no invariance properties. They are thus successful only for invariant sub-patterns. Elements of lines, strokes or line crossings are examples of such subpatterns invariant for many transformations ; thus may be explained the limited success of "Perceptron-like machines" for complex patterns, composed of these simple features.

3. The object of this paper is to propose some "invariant linguistic operators" usually effective for reaching the next level. They apply to two dimensional signals or images. The envisioned geometrical transformations are plane to plane transformations. One is for plane projections, the other for plane similarities.

§II. LINGUISTIC ANALYSIS OPERATORS INVARIANT FOR PLANE PROJECTIONS

II.1. DEFINITIONS

Let the pattern be composed of "primitives" of names $A_0, A_1, A_2, \dots, A_i, \dots, A_n$, to which are associated numerical values $(x_0, y_0), \dots, (x_i, y_i), \dots$. These primitives may be the "features" of the second level. The numerical values x_i, y_i are the coordinates of the "center" of features A_i . Let these "points A_i " be in a plane Π .

Let us establish one of them as privileged, for example. $A_i A_0$ defines a direction. Starting from this direction, let us rotate a vector around A_i in a given manner. This vector will encounter the other points A_j in a certain order.

Noting this order we can obtain an n -word on the alphabet $\{A_j\}$ of the names, starting by A_i :

$$u_i = A_i, A_{j_1}, A_{j_2}, \dots, A_{j_{n-1}}$$

From all the points except A_0 , a word such as u_i may be obtained. Let us name U_X this set of n words of n letters.

A projection from plane Π to plane Π' transforms the word $A_0, A_1, A_2, \dots, A_n$ into $A'_0, A'_1, A'_2, \dots, A'_n$. It is easily verified that the set of n words U'_X , obtained from the new pattern X' by the same algorithm, is derived from U_X by substituting A'_i for A_i .

Thus the operator & implemented by the proposed algorithm is invariant for a plane projection. It is easy to verify that a plane similarity, a rotation for example, does not modify the n words either.

Remarks

1. Let the projection from plane to plane be a "parallel projection", i.e. the center of projection be at infinity. Let G be the center of gravity of the primitive points A_i , which may be weighted by numerical values z_i .

$$\sum_i z_i G^+ A_i = 0$$

G' , the projection of G , is still the center of gravity of the projected points. The quality is invariant for the proposed transformation. Thus for these parallel projections, G may be selected as the "privileged $n + 1$ point" A_0 of an n -points picture.

2. Let A_0 be at infinity. This means that the starting direction of the rotating analysis vector is the same for all the n points A_i . Let us consider the projections from one plane to another keeping at infinity the projection of A_0 . In other words, this means that A_0 is the point at infinity on the intersection line of the two considered planes. For the new plane, the starting direction of the rotating analysis vector is the same for all projected points. It is the projection of the starting direction in the former plane. We will restrict ourselves to such transformations in the proposed application examples.

II.2. PRACTICAL APPLICATION TO IMAGES

Let us take for the common starting direction of the rotating vector, the direction of the x axes (horizontal right). The rotation sense is taken counter clockwise.

Let $\{A_1, A_2, \dots, A_n\}$ be the alphabet of names of the primitives. The partial result obtained from point A_i will be written down as :

$$u_i = \underbrace{A_i, A_k, A_e, \dots, A_q}_{n \text{ elements}}$$

n words such as u_i form an $n \times n$ matrix : the "result U_X ". This result is invariant for the envisioned plane projections.

Any monotone increasing function of the vector angle results in obtaining the u_i . Let q be the size of the computer register in which is stored the angle of $A_i A_j$ with the horizontal axes. Let x and y be the $A_i A_j$ projections on the horizontal and vertical axes. An easily computed two's complement function $\varphi(A_i A_j)$ is :

$$0, \frac{\pi}{2} \dots \left[2 + \frac{|y|}{|x| + |y|} \right] 2^q - 2$$

$$\frac{\pi}{2}, \pi \dots \left[3 + \frac{|y|}{|x| + |y|} \right] 2^q - 2$$

$$\pi, \frac{3\pi}{2} \dots \left[\frac{|y|}{|x| + |y|} \right] 2^q - 2$$

$$\frac{3\pi}{2}, 2\pi \dots \left[1 + \frac{|x|}{|x| + |y|} \right] 2^q - 2$$

Such a function φ covers the available computer numerical range. It orders the angles with a very high discrimination.

N. B. This precision may be superfluous ; as will be seen later, it may be useful to decide that two points A_k, A_e are seen "in line" from a point A_i , in other words that the angles of vectors $A_i A_k$ and $A_i A_e$ with a fixed direction are the same. This is achieved by reducing the function φ precision.

The figures 1-X and 1-Y display images X and Y of 8 primitive points, $A, B, \dots, H$. The corresponding results are given by Fig. 2-X and 2-Y.

The Y image has points "in line". This is recorded in the 8×8 matrix by setting a number after the letter. For instance E and C are in line with A , also H and F ; C and A are in line with E, F and A with H , cf. Fig. 2-Y.

Fig. 1-X

A	B	C	D	E	F	G	H
C	A	H	D	G	F	H	F
B	A	G	H	F	H	F	C
D	C	A	H	G	H	F	F
F	C	A	H	D	F	G	H
E	C	A	D	F	G	H	F
G	F	F	C	D	A	G	H
H	F	C	E	A	D	H	G

Fig. 2-X

Fig. 1-Y

```

A B G E I C 2 H 1 F 2 D
H A G E H F C D
D A B C F G F H
C A B G E H F D
F D A C P E G H
E D C I A 2 B G H F
G F D E C A F H
H D F I A 2 C B E G

```

Fig. 2-Y

Remarks

1. A change in the common angular origin of the rotating analysis vector usually transform U_X : a circular permutation of the $n - 1$ letters in a line after the first may occur.

An image rotation in the plane would introduce such permutations. The considered operator is not rotation invariant. Why then use such an operator rather than the more general one introduced? Because, by comparing two unknown figures, the correspondence between the points A_0 has to be known. In general this is not feasible.

The proposition of taking the center of gravity as the point A_0 is quite interesting, but it restricts the plane projections to parallel projections. On the other hand by using the parallel origin direction vector we will be able to compare figures with different number of primitive points and also to obtain the convex envelope. This would not be possible by using the "center of gravity" origin method. Anyway, the invariance to rotation may be of real value for some problems and this process is worth studying also.

2. If A_0 is at infinity, the image can be scanned from A_0 , in the same sense that the rotating vectors' sense. Here this scanning is made by a horizontal line, the ordinate of which decreases from $+\infty$ to $-\infty$. Thus the n primitive points are put in a certain order. We adopt this order for the first column of the $n \times n$ matrix U_X , which is then uniquely defined ; cf. Fig. 2-X and 2-Y.

II.3. SOME PROPERTIES DEDUCED FROM U_X

The U_X matrix, invariant for plane projections preserving the horizontal direction, is used to recognize large pattern classes.

Already some properties of the analysed image may be deduced from U_X alone. For instance it is possible to state the sense of any three pointed triangle or to decide if any four pointed quadrilateral is convex or not.

The convex envelope

A most interesting property is the ability to find the convex envelope, i.e. the points on

which a convex polygon can be built such that all the other image points are inside this polygon. Let two points $A_k A_e$ be such that :

- In the line starting at A_k , called the A_k line the point immediately after A_k is A_e .
- The A_e line terminates with A_k .

It is easy to demonstrate that A_k and A_e are elements of the convex envelope, and that A_e immediately follows A_k on the envelope for the adopted rotating sense.

The next point A_m on the envelope is such that it is the point following A_k in the A_e line. If A_k is the last point in the A_e line, A_m is the second (circular permutation property).

All the envelope points are obtained through this algorithm, continuing so as to find again the first point A_k .

For instance, going back to the eight primitive points image of Fig. 1-X, the U_X of which is displayed in Fig. 2-X, it is very easy to obtain any U_X corresponding to any partial image, deduced from the original image by suppressing a number of points. Fig. 3 gives the envelopes of images deduced from Fig. 1-X. One line gives the points, the line underneath the convex envelope points. For the total image A B... H, the envelope is C A B G H F.

```

      POINT
      -----
      ENVELOPPE  F A G H F
      POINT
      -----
      ENVELOPPE  F G H F
      POINT  H G
      -----
      ENVELOPPE  C A G H F
      H G F A C
      -----
      ENVELOPPE  C A G H F
      F A G H F
      -----
      ENVELOPPE  F A G H
      C A F
      -----
      ENVELOPPE  F C F

```

Fig. 3

Selected points of Fig. 1-X, followed by the convex envelope points.

Rebuilding the image from U_X

A matrix U_X , being the result of the analysis operator $\mathcal{E}$, is it possible to obtain again the primitive image X from U_X ?

Let us recall that the interest of U_X is not an information reduction from the primitive information of one image, but that a U_X is the same for a class of patterns, modulo a "geometrical" transformation Θ , such as has been envisioned.

Even more, some image "deformations" do not modify U_X . Let us consider a pattern X of n

points, the $n(n-1)$ half-lines joining two points A_i, A_j , the n horizontal half-lines joining a point A_i with the point A_0 at infinity. If none of these n^2 half lines crosses an image point A_k during a "deformation" transform Δ , obviously the result U_X will be unchanged. Such a deformation Δ may alter the primitive pattern shape considerably. Thus any image rebuilt from U_X may differ from the original image X not only by a transformation Θ , but also by a deformation Δ .

The convex envelope of any subpattern may be obtained from U_X . This facility is used to rebuild a pattern X' from U_X . The envelope points are put evenly on a circle in their order. The matrix U_X indicates the regions, defined by a number of straight half-lines, where the other points should be. The set of these other points is ordered in order of decreasing freedom. The first points in this order, thus those with the minimum number of constraints, are put first at the barycenter of the surface where they should be set according to U_X . An impossibility may occur before the set is exhausted. The process is then started again after a random variation of the positions of the points.

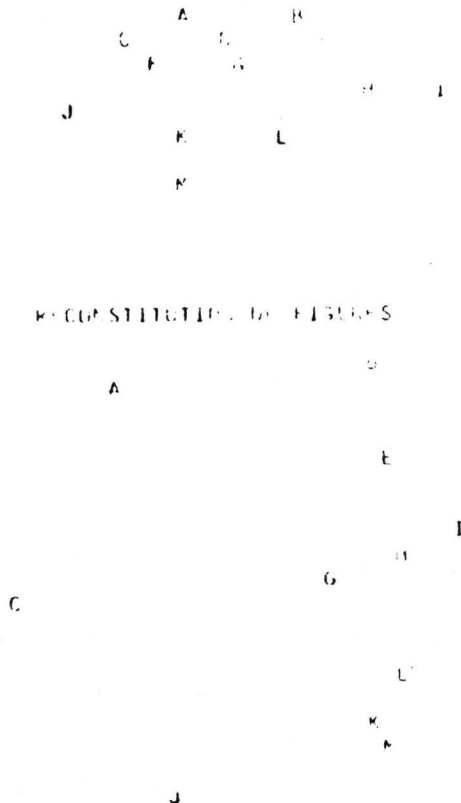


Fig. 4'

Though we are quite convinced that this is not the best idea, successive iterations of this process usually increases the number of points that can be disposed of according to U_Y .

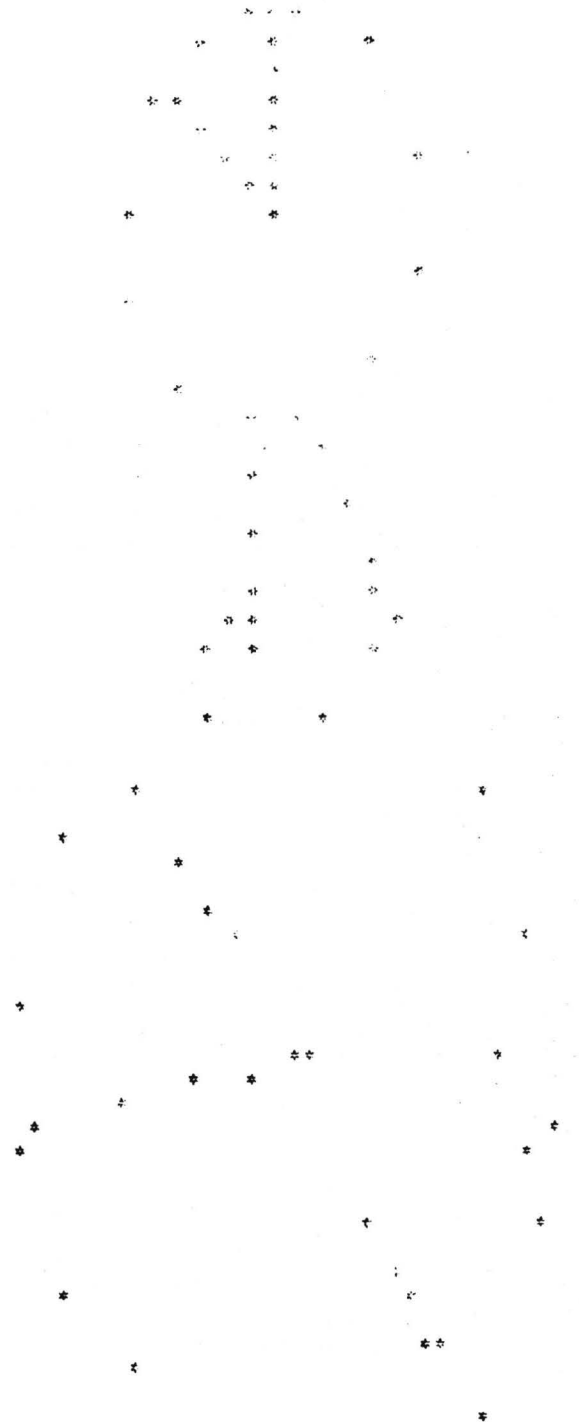


Fig. 5'- Reconstitution of figure 5 rotated of 90°

Fig. 4' shows the reconstitution of the pattern in Fig. 4. All but point **F** are set of a total of thirteen. The convex envelope is A C J M I B. The matrix U_X does not give any indication of the envelope shape, the ignorance of which leads then to an important deformation of the two patterns.

Fig. 5' gives the reconstruction of Fig. 5 pattern from U_X . Out of 38 points, only 30 can be set. This number seems a practical limit.

Though it is interesting to try to rebuild a pattern from U_X , it is not at all the aim of the process, which is principally for pattern identifications. Nevertheless it shows how different two patterns with the same operator attribute U_X may be.

II.4. PATTERN IDENTIFICATION FROM U_X

The identification of two patterns X and Y , having the same number n of points, is quite simple.

Let

$A_1 A_2 \dots A_n$ be the "points" of X ,

$B_1 B_2 \dots B_n$ be the "points" of Y .

The two patterns X and Y are "equivalent", i.e. may be deduced one from the other by a transformation Θ and eventually a deformation Δ , if there exists a bijective application $\Gamma : A_i \rightarrow B_j$

such that U_X is transformed in U_Y , and inversely : $\Gamma^{-1} : B_j \rightarrow A_i$.

The application is defined readily by comparison of the first columns of U_X and U_Y . Then the verification of correspondence is made on the matrix lines. For instance, let X and Y be the patterns of Fig. 1-X and 1-Y.

The application Γ and Γ^{-1} are :

Γ, Γ^{-1}

A	C	B	D	F	E	G	H	(Pat. X)
A	B	D	C	F	E	G	H	(Pat. Y)

The second lines of U_X and U_Y are :

C	A	B	D	G	E	H	F
B	A	G	E	H	F	C	D

Using the defined Γ , a complete correspondence cannot be established between these two lines. We decide that these patterns are not equivalent.

If points are "in line", they may be exchanged during the identification operation.

The discrimination between two patterns of n points is quite high. It depends of course on the angular precision, which may be reduced : two points A_i, A_j of a pattern are "in line" from a point A_k if the angle $(A_i A_k, A_j A_k)$ is smaller than ϵ in absolute value. Thus classes are defined in the A_i line. Points may be interchanged inside a class. A similar process has been tested for the distance operator (cf. paragraph III and

J. C. Simon, A. Checroun and C. Roche (9), (10)) we call it "parenthesizing".

Finding common equivalent subpatterns

Let X and Y be two patterns, $|X|$ and $|Y|$ be the number of primitive points. The analysis operator $\&$ provides U_X and U_Y . From the knowledge of U_X and U_Y is it possible to find two subpatterns Z and Z' such that $Z \subseteq X$ and $Z' \subseteq Y$, $|Z| = |Z'|$, and $\Gamma : U_Z \rightarrow U_{Z'}, \Gamma^{-1} : U_{Z'} \rightarrow U_Z$? In other words, by suppressing a number of points in each pattern is it possible to find two subpatterns satisfying the former correspondence algorithm?

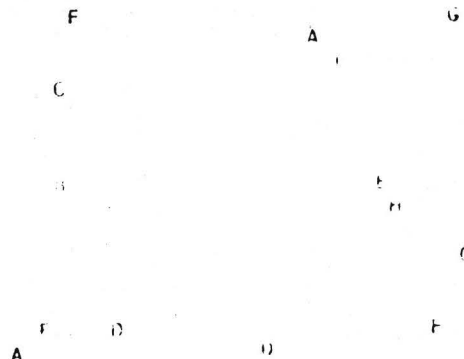


Fig. 6-X

Fig. 6-Y

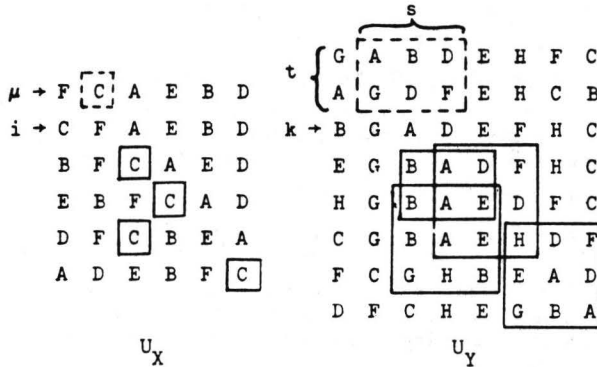
Let us show the process on an example. Let X be a pattern of 6 points, Fig. 6-X; Y be a pattern of 8 points, Fig. 6-Y. Let us examine first if X can be found entirely embedded in Y , (thus $X = Z$).

Referring to U_X and U_Y , the correspondence of certain points is obviously impossible (*). For example, F_X, D_Y .

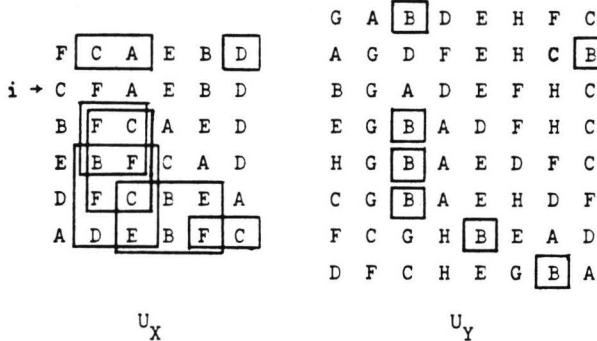
Let us test the possible Y points corresponding to the point C_X . From the first column, only A_Y, B_Y, C_Y may correspond. We shall see that B_Y cannot be a C_X correspondent.

If $\Gamma : C \rightarrow B$, B should be found in all five rectangular domains of U_Y displayed below : the B position in U_Y has to be close to the C position in U_X . In the chosen example B is only in four domains and thus B cannot be a C correspondent by Γ .

(*) When there is some doubt about the points' origin, an index X or Y will state the origin from the X or Y pattern.



Let us examine to see if C may be a B correspondent by Γ^{-1} : $B \rightarrow C$; the point C should be in five of the seven rectangular domains. As the point C is in six of these domains, B may be a Γ^{-1} correspondent of C.



After this example, let us give the general conditions, and let $U_X(i, j)$ be a letter of U_X , $U_Y(k, l)$ be a letter of U_Y .

1. First column correspondence

If $U_X(i, 1) \rightarrow U_Y(k, 1)$

$\text{Sup}(1, |Z| - |X| + i) \leq k \leq \text{inf}(|Y|, |Y| - |Z| + i)$
($|Z|$ is the number of points of the possible partial pattern).

2. An X point exists also in other columns of U_X .

let μ and ν be its coordinates, such that $U_X(i, 1) = U_X(\mu, \nu)$. Each point $U_X(\mu, \nu)$ determines a rectangular domain in U_Y , defined by t and s . The conditions on t are:

$\mu < i$, then $\text{sup}(1, |Z| - |X| + \mu) \leq t$
 $\text{inf}(k - 1, |Y| - |Z| + \mu) \geq t$
 $\mu > i$, then $\text{sup}(k + 1, |Z| - |X| + \mu) \leq t$
 $\text{inf}(|Y|, |Y| - |Z| + \mu) \geq t$

And s must satisfy:

$\text{Sup}(2, |Z| - |X| + \nu) \leq s \leq \text{inf}(|Y|, |Y| - |Z| + \nu)$

$|X| - 1$ such domains are defined in U_Y .

Γ : $U_X(i, 1) \rightarrow U_Y(k, 1)$ is possible if $U_Y(k, 1)$ is at least in $|Z| - 1$ of these domains.

Correspondence matrices M_Z can then be obtained. Let us represent them for the example. A "0" signifies that at least one of the correspondences is impossible, a "1" that both are possible according to the preceding rules.

X \ Y	G	A	B	E	H	C	F	D
F	0	1	1	0	0	0	0	0
C	0	1	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0
E	0	0	0	1	1	0	0	0
D	0	0	0	0	1	1	1	0
A	0	0	0	0	0	0	1	1

M_Z for $|Z| = 6$

M_Z shows that the correspondence is not possible for any Z, Z' such that $|Z| = 6$.

X \ Y	G	A	B	E	H	C	F	D
F	1	1	1	1	0	0	0	0
C	1	1	1	1	0	0	0	0
B	0	0	1	1	1	1	0	0
E	0	0	1	1	1	1	1	0
D	0	0	0	1	1	1	1	1
A	0	0	0	0	1	0	1	1

M_Z for $|Z| = 5$.

Many sets of 5 points in X and Y are candidates for correspondence. Nevertheless this correspondence has to be verified also by the algorithm described for two patterns having the same number of points.

For instance F C B D A of pattern X is equivalent to the Y subpatterns: G B E C D, G A E C D, G B E H D, G A E H D, B E H C F, G B H C D, G A H C D, G B E F D.

Of course the process may be applied to a pattern X itself to find all the equivalent subpatterns $Z \subset X$. For example in the pattern X of Fig. 6-X, F C B D A and F C E D A are equivalent.

Let X and Y be two patterns, chosen at random but having a fixed number of points: $|X| = n_1$, $|Y| = n_2$. It is interesting to use the process to find the possible equivalent subpatterns. A sharp threshold in the number z of equivalent subpattern points is found experimentally. For instance, let $n_1 = 6$, $n_2 = 8$. Table I gives the test results; the first line gives the number z of subpattern points.

The five other lines give the number of equivalent z subpatterns for five different couples of random patterns X, Y .

6	5	4	3	2	1
0	0	120	572	420	48
0	5	144	568	420	48
0	21	249	544	420	48
0	32	266	536	420	48
0	18	180	573	420	48

Table I, $n_1 = 6, n_2 = 8$

If $n_1 = n_2 = 10$, out of 8 couples of random X, Y , no subpattern is found for $z \geq 8$. If $z = 7$, subpatterns are found only for 3 couples out of 8. If $z \leq 7$, many subpatterns are always found.

An approximate value of $\bar{z}$, average of z , is given by

$$\bar{z} = \frac{\begin{bmatrix} |X| \\ |Z| \end{bmatrix} \times \begin{bmatrix} |Y| \\ |Z| \end{bmatrix}}{(|Z| - 1)!}$$

This has to be compared to what we found by linear filtering of sampled noisy images, cf. J. C. Simon and J. Camillerapp (8). A sharp threshold exists under which one may consider that the examined pattern is "noise".

§III. A LINGUISTIC ANALYSIS OPERATOR INVARIANT FOR SIMILARITIES

III.1. DEFINITION

Let $A_1, A_2, \dots, A_n$ be the n "points" of a pattern X as defined in §II.1. From one of these, A_i for instance, an order on the remaining $n - 1$ points may be defined by the euclidian distances between A_i and other points A_j .

Let v_i be the word obtained thus :

$$v_i = A_i, A_{j_1}, A_{j_2}, \dots, A_{j_{n-1}}$$

Let v_X be the set of the n words v_i . This set may be ordered and be represented by an $n \times n$ matrix as before. To introduce an order on the points A_i themselves many processes may be proposed. For instance let G be the center of gravity, the points A_i may be set in the order of their increasing euclidian distance from G . This order, which then defines first column of the matrix V_X , is independant of similarities, as the word v_i itself. But, if certain points are missing, this order may be altered by displacement of the center of gravity. Another useful order on the points A_i is obtained by sweeping a line parallel to a direction ; but then a rotation would induce a circular

permutation on the lines of the matrix V_X .

III.2. COMPARISON WITH A FORMERLY PROPOSED METHOD

This operator has to be compared to what already has been proposed, cf. J. C. Simon and al. (9), (10).

In that method all the distances between two points of a pattern X were ordered giving a unique ordered set D . The reader is referred to the publication (10) for a detailed description of the comparison between two patterns of the same number of points.

By defining a variable distance precision e , classes are introduced on the ordered set D . They are represented by "parenthesis". During the assignment algorithm between two sets D and D' , letters can be taken indifferently inside a class. This method, called "parenthesizing", allows some flexibility in the comparison between two patterns. Let us summarize the results :

If $e = 0$ the assignment is generally impossible ; if e is large enough, the assignment is undetermined. Two situations may arise :

- (a) If $e < e_1$ assignment is impossible,
If $e_1 \leq e \leq e_2$ assignment is unique,
If $e_2 < e$ assignment is undetermined.
- (b) If $e < e_3$ assignment is impossible,
If $e \geq e_3$ assignment is undetermined.

In situation (a), the two patterns are said to be comparable to the precision e_1 .

In situation (b), the two patterns are said to be different.

e_1 is of an order of magnitude smaller than e_2 and e_3 , which are of the same order of magnitude. Again a threshold is found, above which a decision cannot be taken with security.

The "parenthesizing process" can also be introduced for the new operators proposed in this paper. It would introduce some degree of freedom in the comparison, often necessary from the very nature of the pattern "points". The proposed methods are useful after the first level recognition ; then a "point" represents in fact the center of gravity of a detected characteristic feature, thus it may vary to a certain extent.

A drawback of the first proposed method, cf. (10), is that the comparison between two patterns of different numbers of points is not practical. Again, according to the nature of the "points", it seems likely that two compared patterns may have a different number of points.

We believe that the optimum solution will be found by a combination of the two ideas : comparison of two patterns having a different number of points and variable precision, thus parenthesizing. The first performed experiments show that a threshold exists and that it is a function of

the angular or distance precision ϵ and of the number z of the possible common subpattern. Under this threshold, comparison is not performed with security.

A pattern X being given and the ordered set D obtained, it is easy to deduce v_X from D . The knowledge of v_X alone does not allow obtaining D . Nevertheless it is easy to supplement v_X for that purpose.

The reconstruction of a pattern from the knowledge of D or v_X is possible and easier than the reconstruction performed from the §II u_X .

The pattern is obtained modulo a possible similarity. Some deformations may also be introduced; these deformations are usually much smaller than for the distance operators.

From the V_X results the search for common subpatterns may be performed with algorithms similar to those described in §II.4. They will be published later.

§IV. CONCLUSIONS

A generalisation of the analysis operators may be proposed. Let X be a "pattern" made of n "elementary patterns",

$$X = \left\{ \begin{array}{ccccccc} A_1 & \dots & A_i & \dots & A_n \\ x_1 & \dots & x_i & \dots & x_n \\ y_1 & \dots & y_i & \dots & y_n \\ z_1 & \dots & z_i & \dots & z_n \\ \dots & \dots & \dots & \dots & \dots \end{array} \right\}$$

To an elementary pattern of name A_i , qualificatives $x_i, y_i, z_i, \dots$ may be associated. The analysis operator $\mathcal{E}$ delivers n attributes u_i from X . They are obtained with a real function $\varphi(A_j | A_i)$ defined with the A_j and A_i qualificatives. The $n - 1$ elementary pattern names are ordered in a way such that if $\varphi(A_j | A_i) < \varphi(A_k | A_i)$, A_j is before A_k in the word u_i beginning by A_i . A word u_i defines an "order from A_i ". Ordering the elementary patterns names by some other way, an $n \times n$ matrix U_X is finally obtained. This matrix is usually invariant for many geometrical transformations of the image.

Comparison between two patterns X and Y is performed using the matrices U_X and U_Y .

Prospective remarks

(1) From the words u_i , partial order is induced on the set of the elementary pattern couples $\{A_i, A_j\}$. From this order a "clustering hierarchy" in the sense of Jardine or Benzecri is obtained. This hierarchy may be useful for comparing patterns.

(2) Many $\varphi(A_j | A_i)$ functions may be used, particularly distances, satisfying the usual distance criteria. For instance, if the A_i patterns have masses m_i , then

$$d(A_i, A_j) = \frac{d_{ij}^2}{m_i m_j}; \text{ if the } A_i \text{ are straight}$$

lines l_i of relative angle α_{ij} , then

$$d(A_i, A_j) = \frac{d_{ij}^2}{l_i l_j \cos^2 \alpha_{ij}} \text{ etc...}$$

Acknowledgement

The authors wish to acknowledge the participation of graduate students : André, Galland, Gelly, Renard.

REFERENCES

- (1) M. Eden : "On the formalization of Handwriting". Proceeding of Symposia in Applied math, 12, pp. 83-88, Am. Math. Soc., Providence, R. I., 1961.
- (2) M. Eden : "Handwriting and Pattern Recognition". IRE Trans. on Inf. Theory, 8, pp. 160-166, 1963.
- (3) K. S. Fu and P. H. Swain : "On syntactic Pattern Recognition". 3rd Symposia on Computers and Inf. Science, COINS 1969. Bal Harbour, Fla., Dec. 18-20, 1969.
- (4) R. A. Kirsch : "Computer Interpretation of English Text and Picture Patterns" IEEE. Trans. on Elect. Computers, EC-13, pp. 363-376, 1964.
- (5) M. Minsky : "Steps towards Artificial Intelligence". Proc. IRE, 49, # 1, Jan. 1961.
- (6) R. Narasimhan : "Labeling Schemata and Syntactic Descriptions of Pictures", In formation and Control, 7, pp. 151-179, 1964.

- (7) J. L. Pfalz and A. Rosenfeld : "Web Grammars".
Proc. of the Int. Joint Conf. on
Art. Int. (IJCAI) May 1969.
- (8) J. C. Simon and J. Camillerapp : "Recherche
d'une Forme dans un Fond". C. R.
Ac. Sc., Série A, 267, pp. 946-949,
16 Décembre 1968.
- (9) J. C. Simon, A. Checroun and C. Roche : "Com
paraison des Formes Indépendantes
des Déplacements, Homothéties et
petites Déformations". C. R. Ac.
Sc., Séries A, 270, pp. 1,607-1,609
15 Juin 1970.
- (10) J. C. Simon, A. Checroun and C. Roche : "A Me
thod of Comparing two Patterns Inde
pendent of possible Transformations
and small Distorsions". IEEE Sympo
sium on syntactic pattern recogni
tion. Chicago, oct. 1970. To be
published in Pattern Recognition
Journal.