

Recognition of Rotated Characters by Inexact Matching

L. Cinque, S. Di Zenzo, S. Levialdi

Dipartimento di Scienze dell'Informazione, Università "La Sapienza" di Roma
Via Salaria 113, 00198 Roma, Italy

Abstract. In this note we address the recognition of rotated hand-printed characters. We define the sides and the lids of a plane figure and outline an inexact matching process using such features based on the edit distance between circular words of different lengths.

1 Introduction

Numerous object recognition tasks can be seen as a case of inexact matching [1]. In order to solve this problems we must find a correspondence between two relational structures. Given a candidate object to be recognized and a model object called the prototype, we seek a match between such structures which is not necessary a perfect one, but only good enough. Often these structures can be represented by graphs whose nodes stand for subparts and whose arcs represent relationships between the nodes they connect [2]. When matching graphs representing two different objects, an optimal permutation of a matrix that minimizes a similarity distance measure between the graphs must be found.

In order to improve the efficiency of the matching process we must consider: (1) the choice a suitable distance between the structural descriptions to be matched, (2) the reduction of the number of permutations of the candidate structure needed to compute that distance, (3) matching the candidate structure with a subset of the stored prototypes. Regarding point (3), we note that, if a distance measure is available, the search for the best D_k matching D is actually a search for a nearest neighbor (NN). Hence we know that, with suitable preprocessing in storing the D_1, \dots, D_N , it is possible, in principle, to achieve a more efficient search in the recognition process [3].

In this note we focus on points (1) and (2) with reference to a specific problem, namely, the recognition of rotated characters. In this particular OCR problem, current efforts are far below the performances of the standard OCR commercial systems. There are expectations that inexact matching can have a key role to improve the recognition rate. In section 2 we give an overview of the recognition of rotated characters, and in section 3 we define the sides and the lids of a plane figure and outline an inexact matching process using such features.

2 Recognizing rotated characters

Our research is motivated by a computer application that has received much attention over the past few years, namely, the automatic acquisition of line drawings (engineering drawings, land-use and land-register maps, utility maps).

The raster-to-vector conversion of line drawings is a non trivial computer task. Effective techniques for the automatic input of drawings into a database are difficult to implement, and many efforts in this direction over the past twenty years have found limited success. Solutions that meet the requirements of the users to a reasonable extent have been developed only recently [4]. The automatic computer acquisition of line drawings is performed through the basic steps of vectorization of the line-structure and recognition of the text data interspersed in the document. The latter processing step is a particularly difficult OCR problem: indeed, the characters and symbols in a drawing are usually handprinted, and can be however rotated and rescaled. Besides, characters and symbols are often isolated, offering no contextual information; symbols and lines may overlap, and the separation of symbols arranged into strings may be imperfect.

A portion of a digitized drawing is shown in Figure 1. It can be seen that the separation of text data from the rest of the image is usually a critical computing step. In most raster-to-vector conversion systems text is separated from graphics by extracting and classifying the connected components in the image. This procedure fails, however, when symbols touch or overlap lines, something that occurs often when the originals are blurred or degraded.

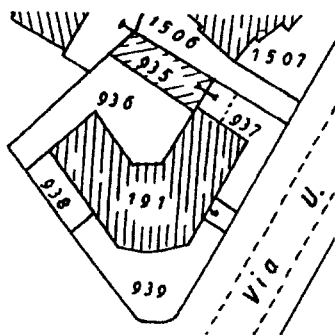


Fig. 1. Portion of a land-register map

We use SEADT[4] a feature-based system, where all the features are invariant under shifts, rotations, and scaling. These invariants are rather obvious requirements, of course: the true criteria in the formation of a feature set has been robustness to distortion and noise. Various reliable features are derived from *lids* and *sides*, as defined in the following. The patterns formed by lids and sides convey much information about the shape of a figure. Figures 2 show some examples of possible lid/side configurations [5].

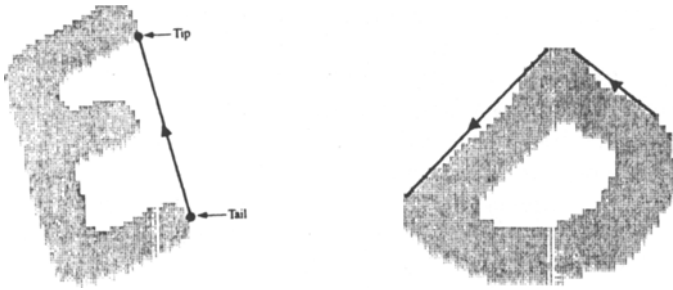


Fig. 2. Example of a lid and sides

SEADT is a rule-based recognition system. That means that each class of symbols has a set of requirements that a shape should satisfy to be considered as a candidate for that class. To have an example, consider symbol T in Figure 4: a very simplified set of requirements that a symbol should satisfy to be considered as a candidate T is the following: (1) there are two lids, (2) neither lids is twice as long as the other, (3) the angle between AB and CD is greater than 110 degrees.

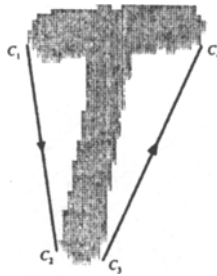


Fig. 3. Shape of T

Each requirement is so formulated that it is satisfied with probability 1 by every shape actually belonging to the class. This strategy allows adding as many requirements as needed for actual discrimination without performance degradation. Our concern in this paper is actually that of implementing techniques of inexact and approximate matching within the framework of the SEADT system. While in this paper we just discuss ideas and proposals, we do have an actual system which seems very suited to the addition of the algorithms discussed below.

3 Inexact matching based on sides and lids

In this section we present and discuss a inexact matching procedure for the OCR of rotated characters. The computational cost is considerably lower. We shall use the notions of side and lid introduced in the cited paper. This section is subdivided in two subsections: in the first we define sides and lids, while in the second we specify a method of inexact matching relying upon those features.

3.1 The lids and sides of a planar figure

Since we are concerned with the recognition of characters of any size, position, and orientation, we need features that do not change value under the similarity transformations of the plane, (motions of the plane that are combinations of shifts, rotations, and scalings). Another requirement for a feature to be of use in this context is the following: it should be as insensitive as possible to certain transformations of the plane such as stretching along one direction and rubber-sheet distortions. These more general motions of the plane are usually present in hand-written characters. Of course, to be of actual use in the classification of shape, a feature should also exhibit certain basic properties: it should have discriminating capability, be rather insensitive to the sampling rate, be computationally cheap, have small variance on each of the subpopulations it helps to discriminate from one another, have high overall variance.

In this section we concentrate on the two features lids and sides. They seem to meet the above wishlist for features to a reasonable extent. However, the use done of these features in that recognizer is somewhat straightforward. It is well known that the number of concavities (or bays) is somewhat unreliable as a feature; indeed, to use it reliably, we should need a threshold to separate noise concavities from true concavities, which obviously does not exist. However, bays can be extremely useful in the recognition of rotated characters: the key idea here is not to use the bays themselves, but their lids.

By a lid of a plane figure we mean a maximal portion of the perimeter of the convex hull of the figure not belonging to the figure itself. The arrow in figure 5 represents a lid. Lids are vectors, each equipped with a tail, a tip, and an orientation. We stipulate that orientation is chosen such that each lid leaves the figure on the left (anticlockwise orientation). To define the sides of a figure, let us first note that the boundary of the convex hull of a figure always consists of a sequence of segments. We normalize the lengths of these segments by dividing them by their sum. Segments whose normalized length is greater than a fixed threshold, typically 0.15, are called sides. A side is a vector, hence it has a tail, a tip, and an orientation. For orientation we stipulate the same convention as for lids.

3.2 Matching strings of lids and sides

Let F be any plane figure. In general, if we traverse the boundary of the convex hull of F , we will encounter certain lids and sides. We may associate to F a

string over a two-letter alphabet $\{s, l\}$ to represent the sequence of sides and lids encountered in the traversal. For example, a string such as sls will denote a sequence composed by a side, followed by a lid, followed by a side. Obviously, the strings sls , ssl , and lss will represent the same sequence of sides and lids.

Pursuing this approach a bit further, we may replace the strings over $\{s, l\}$ with strings of complex numbers in exponential form $re^{i\alpha}$. We stipulate that r represents the length of the vector (side or lid), while the amplitude α represents the angle formed with the subsequent vector ($\alpha = 0$ if there is no subsequent vector).

To distinguish between sides and lids we store $-re^{i\alpha}$ when the vector is a side. This amounts to taking a negative imaginary part for sides (note that the angle formed by a vector (side or lid) with the subsequent vector does not exceed 180 degrees. Thus, we represent the salient features along the boundary of the convex hull by a sequence of complex numbers. As with strings over $\{s, l\}$, two strings that can be obtained from one another by a circular permutation represent the same sequence of features.

Let us now examine how matching can be implemented. In the training phase we will store a string S_k in correspondence of each prototype figure. S_k illustrates the sequence of lids and sides along the border of the convex hull of the k -th figure used to train the recognizer. A similar string S is associated with the query figure F . To reduce to a search for NN (or kNN) we need a notion of distance between strings of complex numbers that possibly can be of different length.

We propose a generalization of the so called *edit distance* between strings over an alphabet. Before we get into the details, however, we find it convenient to note that the number of permutations of the query string S needed to find the best match is not greater than to the length of S . Indeed, we only have to permute S circularly.

The edit distance is connected with the string-to-string correction problem. This problem is to determine the distance between two strings as measured by the minimum cost sequence of edit operations needed to change one string into the other. The edit operations allow changing one symbol in a string, deleting one symbol from a string, inserting a symbol into a string. The cost of each of these edit operations is set equal to a constant C (usually, $C = 1$). There are various algorithms that compute the edit distance between two strings in time proportional to the product of the lengths of the strings [6]. Fortunately enough, a generalization of the edit distance to circular strings has already been studied, and we also have an algorithm that computes the distance between circular strings in $O(nm \log m)$ time [7].

To generalize the edit distance to strings of numbers we simply replace the constant C with the absolute value of the number that is inserted or deleted, or the absolute value of the difference of the two numbers, when a number is replaced with another. It is straightforward to show that the quantity so computed satisfies the axioms of a distance measure. The overall complexity of this matching is thus $O(n^2 \log n)$, where n is the average length of our strings.

References

1. L. G. Shapiro and R. M. Haralick, "Structural descriptions and inexact matching", *IEEE Trans. on Pattern. Analysis. Machine. Intell.*, **5**, pp. 504-519, (1981).
2. A. Sanfeliu and K. S. Fu, "A distance measure between attributed relational graphs for pattern recognition", *IEEE Trans. on System, Man and Cybernetics*, **13**, pp. 353-362, (1983).
3. F. P. Preparata and M. I. Shamos, *Computational Geometry — An Introduction*, Springer-Verlag, 1985.
4. L. Boatto, V. Consorti, M. Del Buono, S. Di Zenzo, V. Eramo, A. Esposito, F. Malcarne, M. Meucci, A. Morelli, M. Mosciatti, S. Scarci, M. Tucci, "An Interpretation System for Land Register Maps," *IEEE Computer*. **25**, 25-33 (1992).
5. S. Di Zenzo, M. Del Buono, M. Meucci, A. Spirito, "On the recognition of hand-printed characters of any size, position, and orientation," *IBM J. Res. Develop.* **36**, 487-501 (1992).
6. R. Wagner and M. Fisher, "The string-to-string correction problem", *Journal of the ACM.*, **1**, pp. 168-173, (1974).
7. M. Maes, "On a cyclic string-to-string correction problem", *Information Processing Letters*, **35**, pp. 73-78, (1990).