MVA'94 IAPR Workshop on Machine Vision Applications Dec. 13-15, 1994, Kawasaki

RECOGNITION OF CARTOGRAPHIC SYMBOLS

Sushil Bhattacharjee Gladys Monagan

Institut für Informationssysteme
Swiss Federal Institute of Technology (ETH)
ETH-Zentrum, CH-8092 Zurich, Switzerland

ABSTRACT

A hybrid (statistical/structural) approach is presented, for scale- and orientation-invariant recognition of multi-component cartographic symbols. A decision-tree classifier (DTC) is used to identify the shapes of the individual components of a symbol. Structural matching is then used to determine the type of symbol under consideration.

INTRODUCTION

Machine-interpretation of cartographic maps has come to occupy an important place in the burgeoning document-image-processing industry. Several comprehensive collections of papers on this topic are now available [1]. Cartographers often use predefined symbols to convey such 'meanings' associated with logical structures represented in maps. Recognition of cartographic symbols is, therefore, an important aspect of any map-interpretation system.

In this paper, we describe a method for recognizing cartographic symbols that has been developed for processing digital images of land-registry maps in Switzerland. These maps are basically line drawings which identify the various regions of an urban neighborhood. The proposed method for symbol recognition operates on bilevel images. It is independent of the size and orientation of the hand-drawn symbols, and is also independent of the scanning resolution of the input image.

Figure 1 presents an overview of the symbol-recognition approach proposed here. The different stages of the flowchart shown

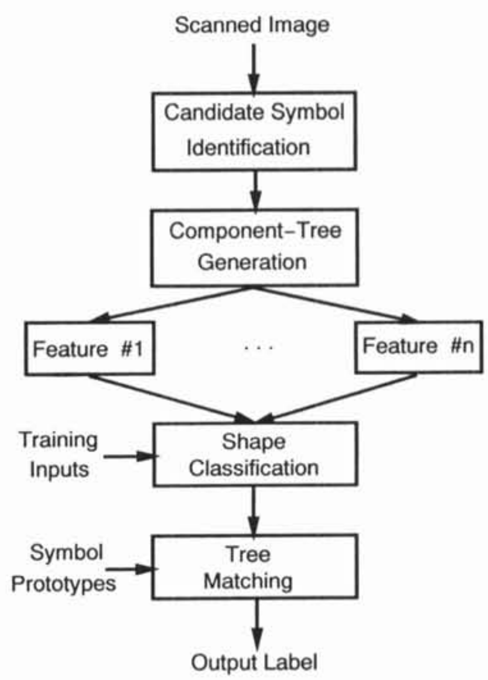

Figure 1: Overview of the symbol recognition process. A binary decision-tree classifier is used for shape classification. The shape features used are designed to be independent of scale and orientation.

in Figure 1 are discussed, in that order, in the following sections. Some closing comments are offered, after a discussion of experimental results.

PREPROCESSING

Regions that contain potential candidates for symbols are first selected from the input image. This selection is performed on the basis of certain general criteria involving the size and proximity of connected regions of black pixels. A rectangular subimage enclosing each selected region is used for further processing.

In the general case, cartographic symbols have several distinct components. A hierarchical connected-component representation [2], referred to here as the *component-tree*, of the input symbol-subimage is generated. Each node of the component-tree represents a connected region in the subimage. The hierarchical structure of the component-tree preserves the relationships of neighborhood and inclusion among the individual components of the symbol. This component-tree is then processed to recognize the corresponding symbol. First, the individual components are identified using a shape recognition approach. A simple tree-matching procedure then identifies one of the known symbol-prototypes that matches the structure of the input symbol.

SHAPE FEATURES

A set of shape features is used to identify each individual component in the component-tree. These shape features were recently used by Di Zenzo *et al.* [5] for optical character recognition (OCR). The shape features are derived mainly from the convex-hull of the shape (component) in question. The *convex-hull* of a shape is the smallest polygon that completely encloses the shape, and it can be computed from its outer contour. Following the terminology of Di Zenzo *et al.* [5], we use the term *bay* to refer to a significant convex-deficiency. (A convex-deficiency is considered to be significant if its area is greater than a predetermined fraction of the total area within the convex-hull.) The shape features used in this study are briefly described below:

1. Color of Centroid: If the centroid of the convex-hull belongs to the shape itself, its *color* is '1', otherwise it is '0'.
2. Circularity: The *circularity* of a shape is given by the ratio A/C, where A is the area of the shape and C is the area of the smallest circumscribing circle centered at the centroid of its convex-hull.
3. Number of Holes: A substantial region of background color, completely enclosed within the shape, is called a *hole*.
4. Number of Sides: Every segment of the convex-hull polygon is a potential *side*. The normalized length (length of the segment divided by that of the entire convex-hull) is first computed for each segment of the convex-hull. Segments having a normalized length greater than some predefined threshold (some fraction of unity) are considered valid sides. Every side is assigned a direction such that, while traversing the side from *tail* to *head*, the shape in question lies to the left of the side. This definition makes a side invariant to scale and orientation, and robust to noise.
5. Number of Lids: A *lid* is defined as a side which 'covers' a bay of significant area.

   Sides and lids, as defined above, are vectors, and the following operations can be defined on these vectors.

   - Nearness: Two vectors are said to be *near* each other if the shortest normalized distance, along the convex-hull, between them is smaller than a predetermined threshold, T_1. Similarly, two vectors are said to be *far* from each other if the shortest normalized distance between their tips is greater than some threshold, T_2.
   - Cooperation/Competition: Two vectors are said to *cooperate* if their scalar product is positive. If the scalar product of two vectors is negative, they are said to *compete*.
   - Twisted pair: Two *competing* vectors that are *far* from each other are said to form a *twisted pair*.
   - Consecutiveness: Two vectors are said to be *consecutive* if there is no other vector between them.

   The following higher level shape features can now be described:

6. Side-torsion and Lid-torsion: The number of twisted pairs composed of the sides of a shape, is called the *side-torsion* of the shape. *Lid-torsion* is defined similarly for lids.
7. Side-chain-length and Lid-chain-length: A *chain* consists of a sequence of vectors (of the same type, side or lid), such that each vector, except the first one, is near and consecutive to the previous one. The length of the chain is given by the number of vectors in the chain.
8. Side-cycle-length and lid-cycle-length: A chain is called a *cycle* if the first and last vectors in the chain are near and consecutive to each other. If a chain is not a cycle then its *cycle-length* is defined to be zero.

SHAPE RECOGNITION

The shape features are arranged in an ordered set, called a *pattern*. Di Zenzo *et al.* [5] use a classification program consisting of several blocks of code, each specifying the expected features-values of one of the known shapes. Our recognition-scheme differs from theirs since we use an automatically designed classifier to assign a *shape-label* to each shape, depending on its pattern. The shape features described above are not well-suited for use with statistical-parametric classifiers. In defining the shape features, the two most important criteria have been invariance to variations in scale and orientation, and stability under noise conditions. Therefore, we expect the features to show very little variability within a class. This indicates that it should be easy to design a simple binary decision-tree classifier (DTC) for the recognition task. Such a classifier can be thought of as a binary tree in which each non-terminal node represents a decision involving the comparison of the value of a specific feature in the pattern to a predetermined threshold. If the feature-value is greater than the threshold, the decision process descends along the 'right' path, otherwise the node along the 'left' path is selected next. This process is repeated until a terminal node of the decision-tree is reached. At this point a label is assigned to the input pattern.

Supervised training of this classifier involves the design of the binary-decision-tree, using training patterns from each known shape-class. Several procedures for automatically designing a DTC have been proposed in the pattern recognition and machine learning literature [3]. The DTC used here is designed using the algorithm proposed by Sethi and Sarvarayudu [4] which is based on the concept of *average mutual information*. (In this limited space, it is not possible to discuss the details of the training algorithm. Kindly refer to [4] for a complete description of it.) After the DTC has been designed using a set of training patterns, each leaf (terminal node) of the DTC is exclusively associated with a subset of the training patterns. Sethi and Sarvarayudu [4] propose the following static labeling criterion when an input pattern reaches a given leaf of the DTC: determine the class which has the maximum number of representatives in the set of training patterns associated the leaf (i.e., determine the 'majority class' at the leaf). Assign the label of this class to every input pattern that reaches this leaf. In contrast, we use a dynamic labeling scheme. Among the training patterns associated with the leaf, the pattern closest to the input pattern is determined, and its label is assigned to the input pattern. The distance between two patterns is expressed as a cosine measure, and therefore lies between 0 and 1. The distance to the nearest training pattern is also used as a measure of confidence of the resultant labeling, and can be used to implement a reject option.

TREE MATCHING

Following the shape-classification step, a complete description of the composite symbol is available in terms of the shape-labels of the individual components, and their inter-relationships. The structural descriptions of all the symbols in an application domain are available *a priori*, and these are stored in *prototype-tree* structures. The labeled component-tree of the input symbol-subimage is compared with each prototype-tree, in turn, until a match is found. Rotational invariance in the recognition is achieved by not enforcing any order among the sibling-nodes in the component-tree. The symbol represented by the component-tree is assigned the same label as the matching prototype-tree.

RESULTS

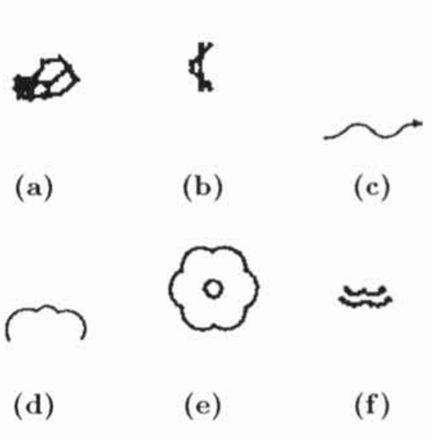

Figure 2: Examples of symbols found in Swiss land-registry maps: (a) solitary rock; (b) vineyard; (c) direction of flow; (d) wooded region; (e) a single tree; (f) open water region.

Shape	1	2	3	4	5	6	7	8	9
a	3	0.278	1	2	0	2	1	0	1
b	1	0.198	1	3	1	1	3	1	1
c	0	0.037	1	4	1	4	3	0	3
d	0	0.053	0	3	1	2	1	0	1
e1	1	0.437	0	2	1	1	0	0	0
e2	1	0.114	0	1	0	1	0	0	0
f1	0	0.130	1	2	0	1	2	0	1
f2	0	0.166	1	2	0	1	2	0	1

Table 1: Each row shows the nine feature-values for a given shape. The letters in the beginning of each row identify the shape with reference to Figure 2. Shapes **e1** and **e2** are the inner and outer components, respectively, of the symbol shown in Figure 2(**e**). Similarly, shapes **f1** and **f2** are, respectively, the lower and upper components, of the symbol of Figure 2(**f**). The features computed are: 1. number of holes; 2. circularity; 3. color of centroid; 4. number of sides; 5. side torsion; 6. side chain length; 7. number of lids; 8. lid torsion; and 9. lid chain length.

The symbol recognition algorithm described in the previous sections is part of a map-interpretation system being developed primarily to process land-registry maps of Switzerland. Figure 2 shows some images of (hand-drawn) symbols used in our maps. A few other symbols that are used in these maps are not included here. Figures 2(e) and 2(f) illustrate the two inter-component relationships (containment and neighborhood) that are possible between components of a multi-component symbol.

Table 1 shows the shape feature-values computed for the different shapes shown in Figure 2. In computing these features, a threshold of 0.1 was used to establish the validity of sides and lids. That is, segments of the convex-hull polygon that were at least as long as 10% of the entire length of the convex-hull were accepted as valid sides. The DTC for shape-classification was designed using four randomly chosen training patterns per shape-class. In our tests, the symbol-subimages to be recognized were not used in the training process. Perfect recognition was achieved at both levels (shape-recognition of the individual components, as well as complete recognition of the composite symbols).

COMMENTS

A hybrid approach for recognizing cartographic symbols has been presented above. Symbols with multiple components are represented in a hierarchical component-tree which preserves the inter-component relationships. First, an automatically designed DTC is used to label the individual components that make up a symbol, by identifying their shapes. The shape-features used here are invariant to scaling and rotation, and are very stable in the presence of noise. They are highly intuitive, and the feature-set can be easily extended if required. The DTC selects only those features that are best suited for partitioning the feature-space appropriately, for the given set of known shapes. In our experiments the decision tree had a height of only four levels. This demonstrates that the shape-classes are fairly well clustered in the feature-space and therefore the features used are quite powerful for shape-discrimination. Shape classification is followed by a structural matching step, in which the labelled component-tree is compared with a series of prototype-trees to select the prototype-tree that matches the input component-tree. In our implementation, a strict match is required in order to identify the complete symbol. In other situations, partial tree-matching can also be used. This could be particularly useful in correcting mistakes made in the previous shape-classification step.

Acknowledgments: The authors thank the financial support of the Swiss Federal Commission for the Advancement of Scientific Research (KWF), project 2540.1, and of the Aargauisches Elektrizitätswerk (AEW). AEW also provided us with the test data used here.

REFERENCES

[1] H. S. Baird, H. Bunke, and K. Yamamoto, editors. *Structured Document Image Analysis*. Springer-Verlag, 1992.

[2] E. Mandler and M.F. Oberländer. One-Pass Encoding of Connected Components in Multi-Valued Images. In *Proc. 10th Int. Conf. Pat. Rec., Atlantic City, USA, 16-21 June*, volume 2, pages 64–69, 1990.

[3] S. R. Safavian and D. A. Landgrebe. A Survey of Decision Tree Classifier Methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, May/June 1991.

[4] I. K. Sethi and G. P. R. Sarvarayudu. Hierarchical Classifier Design Using Mutual Information. *IEEE Trans. on PAMI*, 4(4):441–445, July 1982.

[5] S. Di Zenzo, M. Del Buono, M. Meucci, and A. Spirito. Optical Recognition of Hand-Printed Characters of any Size, Position, and Orientation. *IBM Jou. of Res. and Dev.*, 36(3):487–500, May 1992.