MAP SYMBOL RECOGNITION USING DIRECTED HAUSDORFF DISTANCE AND A NEURAL NETWORK CLASSIFIER†

Eric Reiher*, Fady Said**, Ying Li*, and Ching Suen**
* Centre de recherche informatique de Montréal
1801 McGill College Avenue, Suite 800
Montréal, Québec, Canada, H3A 2N4
** Centre for Pattern Recognition and Machine Intelligence
1550 de Maisonneuve Blvd West, Suite GM-606
Montréal, Québec, Canada, H3G 1M8

Commission III, Theory and Algorithms

KEY WORDS: GIS, Cartography, Vision, Neural Networks, Pattern Recognition

ABSTRACT

A method for map symbol recognition is presented in this paper. Our objective in developing this recognition method is to make recognition efficient, robust and near perfect for handling very large maps with many symbols of different scales and orientations. Our method first utilizes the directed Hausdorff distance as a measure of similarity for selecting possible candidates of user defined models of symbols. This selection will collect as many candidates as possible, in order not to miss any symbols. Neural networks are then utilized to eliminate the false positives among those candidates. Implementation details and experiment results are presented.

1. INTRODUCTION

Currently, we are undertaking a pre-competitive research project on converting paper maps to electronic format. A potential application of our research result is to integrate utility maps, hand drawn several decades ago, into current geographic information systems (GIS). Integrating paper maps into a GIS is a problem of central importance as GIS become more widely used for representing spatial data. On paper maps, graphical symbols are used to indicate the location of various sites. Currently there are research projects and commercial products on vectorizing map images. However, it remains a research objective to automatically locate and recognize user defined symbols, given by a model, from images of scanned maps without assumptions and constraints about the symbols. It is not a trivial task because the symbols in the image can be touching, overlapping, of different scales and orientations from the model, or modified by noise such as that introduced during scanning (Fig. 1). The task becomes even harder when near perfect recognition is required with a speed acceptable for commercial applications. The conversion process consists of scanning the maps, extracting all the information present in each map and storing it in a fashion that it can be accessed and manipulated by target applications. The quality and cost of this convention process are determining factors for a company's decision regarding the adoption of the process in its commercial practice. This process is continuous as new areas or new maps need to be incorporated in the GIS applications. Thus efficiency is also an important determining factor.

This paper presents a recognition algorithm based on Hausdorff distance and neural networks, where our main contribution is to make the recognition efficient and robust for handling very large maps and many symbols of different scales and orientations. Section 2 reviews previous work on symbol recognition. Section 3 outlines our approach. Section 4 explains implementation details. Section 5 presents some experimental results and Section 6 concludes.

† This research is partially sponsered by M3i Systems Inc., Longueuil, Québec, Canada.

2. REVIEW OF PREVIOUS WORK

Our decision of using the Hausdorff distance for symbol recognition is based on our substantial survey of previous work on symbol recognition and shape detection in general. Among more than a dozen methods studied, the Hausdorff approach stands out as the most suitable for our objectives in terms of robustness, efficiency, correctness and scope. This section briefly reviews the previous work related to symbol recognition. The next section introduces Hausdorff distance and our approach.

The generalized Hough transform is probably one of the oldest and best known methods to detect any arbitrary shape, of any orientation in an image (Ballard, 1981). Although this approach could be used to perform map symbol recognition, it seldom is because it usually generates a lot of false positives when applied to complex images to detect complex shapes (Grimson, 1986). The computation can also become quite expensive, as its complexity and memory requirements increase with the complexity of the shapes, and their orientation. The memory required can quickly become unmanageable.

A lot of existing symbol recognition systems use invariant features for shape recognition. In (Wu, 1986), Wu and Stark present an algorithm to recognize an object in an image regardless of the object's size and orientation. They mainly use two features: a circular harmonic function that is rotation invariant and the Mellin transform which is scale invariant. Other popular features are the UNL Fourier features (Rauber, 1994), Wavelet transform (Antonini, 1992), moment invariants (Reeves, 1988 and Reiss, 1991) and Fourier descriptors (Jun, 1986 and Lin, 1987).

A set of features is usually used to classify symbols. Since the instances of each symbol may vary, several instances of each symbol are required in order to build a representative training set that will produce reasonable recognition rates. This training set is constructed dynamically with the help of the user. At first, the user will interactively correct the recognition when necessary. All the symbols identified and corrected by the user will help consolidate the classification. Once the user is satisfied with the classification, the system can then run stand-alone without user interaction.

A good example is the system by Samet and Soffer (Samet, 1994). It uses the (user-located) legend to initialize a table of models to be found in the rest of the image. Then the system divides an image into its constituent elements using a connected component labeling algorithm. For each region a set of features is computed. The system uses global and local shape descriptors identified as features that best discriminate between geographic symbols (Levine, 1982). These features are invariant to scale, orientation and translation.

In general, however, a feature based approach is hard to implement when symbols can overlap or touch each other. Segmentation must then be performed before feature calculation and the resulting problems can rapidly become very complex. In (Gorman, 1988), Gorman *et al.* use a dynamic programming method to attack the segmentation problem. Different segmentation possibilities are tried depending on what is most likely to be recognized so far. This method is mostly used for recognizing complex shapes. For example, if one wishes to distinguish an image of an airplane from that of a train and only a wing with a reactor is showing, the system will easily recognize the object as an airplane. However, it is not clear if the method would yield good results if used to recognize very small and similar symbols.

Several other authors have suggested approaches using almost only heuristics. Kasturi has worked on the problem of separating text from graphics (Kasturi, 1986, Fletcher, 1988 and Kasturi, 1990) and also on locating simple symbols consisting mainly of small loops like rectangles, circles, etc.

Yu *et al.* (Yu, 1994) and Shimotsujia (Shimotsuji, 1990) rely on simple heuristics to perform segmentation. Based on features, shapes like lines, circles and arcs are recognized and grouped. A simple matcher is used to perform recognition. The user can input any type of symbols.

Okazaki *et al.* (Okazaki, 1988) described a complete system for electric circuit symbol recognition. Most symbols in an electrical circuit diagram contain at least a closed loop. An interesting feature about this system is its hybrid approach. It first computes a set of features to isolate loops of interest. Then it uses another set of features to recognize the exact symbol. This recognition is mediated with a pattern matching attempt on the same symbol. The two recognition schema collaborate in order to find the right symbol.

Heuristic methods have several problems. First, they are not general. They can be used to find only a small set of symbols. Their major disadvantage is that they are very sensitive to noise. If a symbol is incomplete, blurred, touching or overlapping another one, it might be missed.

3. OUR APPROACH

The Hausdorff distance measures the degree of mismatch between two sets of points by measuring the distance of a point of one set that is farthest away from any point of the other, and vice versa. Formally, the directed Hausdorff distance h between a model M and an image I at a specific point in the image is:

$$h(M, I) = \max_{m \in M}\left(\min_{i \in I}(distance(m, i))\right) \quad (1)$$

The Hausdorff distance H is defined as:

$$H(M, I) = max(h((M, I), h(I, M))) \quad (2)$$

This distance can be used to determine the degree of mismatch between two objects that are superimposed on one another.

One of the most interesting properties of the Hausdorff distance is that it obeys metric properties. The function is positive everywhere and has the properties of identity, symmetry and triangle inequality. These properties correspond to our intuitive notions of shape resemblance, namely, that a shape is identical only to itself (and not one having identical features), the order of comparison of two shapes does not matter, and two shapes that are highly dissimilar cannot both be similar to some third shape. This final property (the triangle inequality) is particularly important in pattern matching applications in which several stored model shapes are compared with an unknown shape. Thus, two highly dissimilar models cannot both be similar to an unknown symbol.

However, the distance threshold alone is not good enough to discriminate between good and bad matches. When using a small (tight) distance threshold, a lot of poorly drawn symbols can be missed. This is undesirable in some applications. On the other hand, a lot of false alerts can occur when using a threshold large enough not to miss any symbols. We have added a second threshold in order to lower the number of false alerts. The percentage of directly matching pixels from the model on the image is also computed. A threshold on this percentage helps to reject in some cases up to 75% of false matches.

The Hausdorff distance is not sensitive to symbols touching or overlapping each other. It can thus naturally be used to first locate possible candidates. As an extra verification step, any feature can be computed for recognition. In our approach, we have chosen to use the Hausdorff distance in conjunction with a multi-layer neural network for recognition. The found candidates are passed through the network for user-guided training or for recognition once trained.

4. IMPLEMENTATION

4.1 Hausdorff distance

The Hausdorff distance is calculated for every symbol according to Huttenlocher's approach (Huttenlocher, 1993). Optimizations to prune the search area are described. For example, suppose the Hausdorff distance between a symbol and an image at a specific point (x, y) is (d). When this distance is greater than the symbol's threshold (t), then a region of size $(d - t)$ can be pruned (not searched further) surrounding the point (x, y). This is due to the fact that the Hausdorff distance cannot decrease more rapidly than by 1. The only difference in our implementation, is that we start by looking for symbols at the center of the image. We then proceed in the image in a dichotomous way, searching in the middle of yet unsearched region. Huttenlocher starts the search at the top left corner and proceeds left to right, top to bottom. So he can only prune smaller regions. Our method has contributed to a 20% decrease in processing time.

Usually, when a match is found, the Hausdorff matrix will contain not one but a whole region of positions that are below the user's threshold. For example, a small filled circle can be placed at various locations inside a larger one. Our algorithm will choose the position where there is a maximum number of pixels matching, while surrounding matches will be eliminated. When a candidate is found, its position, scale and orientation are passed to the neural network.

4.2 Neural network classifiers

A set of neural network classifiers are used to validate or reject the candidates found by the Hausdorff method. One network per symbol is used for training and validation. The regions of interest to be processed are all presented in the same orientation and scaled down to a 20x20 matrix that constitutes the input layer of the network. There is only one hidden layer that is fully connected to the input and to the output layer. The signal produced by the output neuron is considered to be a measure of confidence that the candidate presented to the network is a symbol or whether it is just a false positive. The output is a floating point number between 0 and 1. The closer it is to 1, the more confident we are that the candidate is a symbol.

The networks are trained using the error backpropagation algorithm (Rumelhart, 1986, LeCun, 1990 and Krzyzak, 1990) with a momentum term and an adaptive learning rate, to which, we have added our own modifications (Said, 1995). While training the different networks over their respective symbols, each network, as it converges, chooses its own optimal parametric values (learning rate, momentum rate, and the sigmoidal activation function's parameter) and the number of hidden neurons, within a user specified range. While training, the error function slope is examined to decide whether to decrease or increase the current number of the hidden neurons, and change the parameter' values. This technique helps in not falling into a local minima along the error function's graph, however, on the average, it results in more iterations (epoch).

Training the system to recognize a new symbol involves two steps. First, the user is asked to build a training set for the symbol, with the assistance of the system. Instances are located by the Hausdorff method, and the user flags each candidate as being valid or not. Second, the network attached to this newly created symbol is trained to differentiate between the symbol and its false positives. The network is then ready to be used in recall mode for automatic recognition.

4.3 Advantages of our approach

Combining the Hausdorff distance with neural networks has several advantages:

- The Hausdorff distance performs segmentation. It has no problem when symbols are touching or overlapping. It also rotates all symbols to the same orientation for easier use by the neural networks.
- The Hausdorff distance acts as a pre-filter. Not all symbols in a map will be passed to a symbol's neural network, only the ones that are close enough. This reduces the training set and augments accuracy.
- The advantage of using one neural net per symbol is to have independent learning and recognition. When a new symbol is added, training on others is not affected.

4.4 Selection

The recognition of each symbol is performed independently. Multiple candidates could thus appear at the same location. A decision must be taken as to which symbol is the best candidate.

All the found symbols are first sorted, the ones with the most matching pixels appearing first. Then the symbols are removed from the image one by one. However, before removing a symbol, we compute the number of remaining black pixels it overlaps. If that number still exceeds the user threshold, we remove it and add it to a list of recognized symbols. Otherwise, the symbol is discarded. When two symbols match at the same (or close) location, this method selects the symbol matching the most pixels first. If there are enough remaining pixels (a rare case), then the other symbol can also be selected.

5. EXPERIMENT RESULTS

We used a 7000x5000 pixel map for training and for adjusting the thresholds. Another map of similar size was used for computing the results. It takes around one minute to process one symbol in one orientation. Table 1 presents results using only the Hausdorff distance. Unfortunately, some symbols were missed. Most misses are caused by very badly drawn symbols. Most come from regions where symbols are hand drawn in very crowded spaces. The problem symbols are usually very small and very badly written. Such symbols were not present in the training map.

Table 2 shows the recognition results when trained neural networks are also used, with a measure of confidence of 0.85. When the output of the neural network is greater than 0.85, the candidate is accepted. This very high measure of confidence is better used when the goal is to accept only the most probable symbols. Some might be missed but no candidates should be accepted by error as is reflected by our results. Only the filled circle is not correctly recognized. This is due to the fact that the test image contains a lot of false positive instances that were not present in the training image. For this symbol, more training would thus be necessary.

Table 3 shows the result when using trained neural networks with a measure of confidence of 0.5, where candidates will be accepted when there is more confidence that they are symbols than false positives. Here only a few symbols are missed, and only a few false positives remain.

Table 4 shows the result when using trained neural networks with a measure of confidence of 0.15, where candidates are accepted when there is no compelling evidence that the symbol should be rejected. When comparing to the symbols missed by the Hausdorff distance, almost no misses are caused by the neural networks. But more false positives are present. For our application, this is the best threshold to use, because it is easier for a user to remove false positives than to search for missed symbols. Again, these results confirm that more training is necessary for the filled circle. The six extra misses for the filled triangle all come from larger triangles not present in the training set. Again, more training would fix this problem.

Figure 1 shows a 1200x400 part of the test map. Figure 2 shows the recognition of ellipses in Figure 1, as produced by the Hausdorff distance. All of them are recognized, but one false positive was produced near the lower right corner. The false positive is later correctly removed by the symbol's neural network.

As expected, the Hausdorff distance produces a lot of false positives. But after neural network filtering, the recognition results are acceptable for user corrections. We expect even better results when training is performed on more than one map (and thresholds are adjusted not to miss any symbol) making user corrections simpler. For sets of maps that are better drawn, over 98% recognition rates have been accomplished.

For the image shown in Figure 1, we get perfect recognition. It contains a lot of different symbols, in different orientation and scale, touching and overlapping each other. All 83 instances of the 6 symbols shown in the tables are correctly recognized in around 30 seconds.

6. CONCLUSION

Our strategy for achieving near perfect recognition is to first generate results that have near zero miss and then reduce the (numerous) false positives to an amount that can be quickly handled by a human operator. Symbol recognition based on Hausdorff distance combined with individually trained neural networks results in accurate recognition.

Our objectives for the near future are as follows. First, we would augment the functionality of the system with various additions such as facilities for the user to specify constraints on the symbols and their surroundings. Second, we will study maps for new application areas (such as navigation and terrain modeling) for which a knowledge-based component of the system will become very important since information needs to be inferred by the system from the extracted symbol information. Finally, we will extend our work to use color images as input instead of only black and white ones.

Table 1: Hausdorff

Symbol	Correct	Miss	False positive
	5	0	39
	17	1	34
	30	2	145
	49	3	129
	127	26	241
	29	1	178

Table 2: Hausdorff + Neural Network (0.85)

Symbol	Correct	Miss	False positive
✚	5	0	0
⧩	14	4	0
⧨	27	5	0
⬭	48	4	0
●	119	34	39
►	15	15	0

Table 3: Hausdorff + Neural Network (0.50)

Symbol	Correct	Miss	False positive
✚	5	0	1
⧩	15	3	1
⧨	30	2	0
⬭	48	4	0
●	122	31	56
►	17	13	1

Table 4: Hausdorff + Neural Network (0.15)

Symbol	Correct	Miss	False positive
✚	5	0	6
⧩	17	1	5
⧨	30	2	0
⬭	49	3	5
●	125	28	62
►	23	7	6

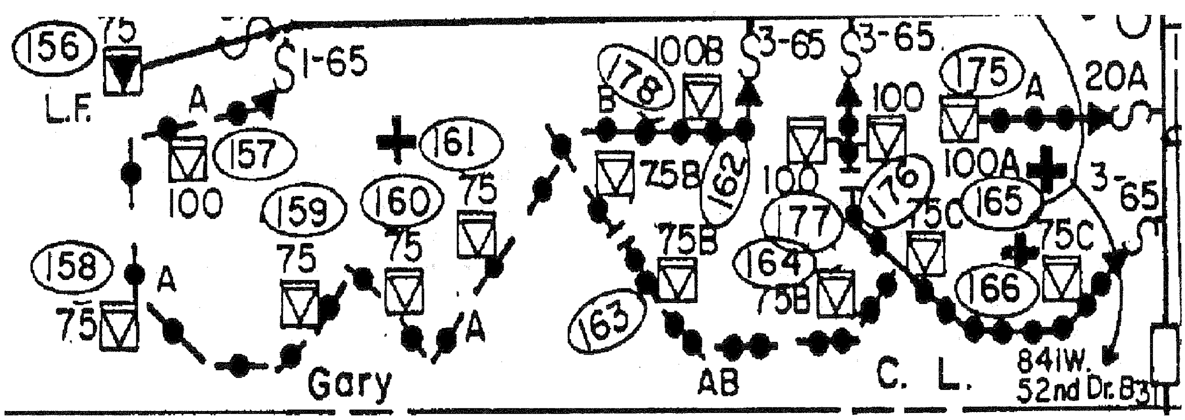

Figure 1: Part of the test image

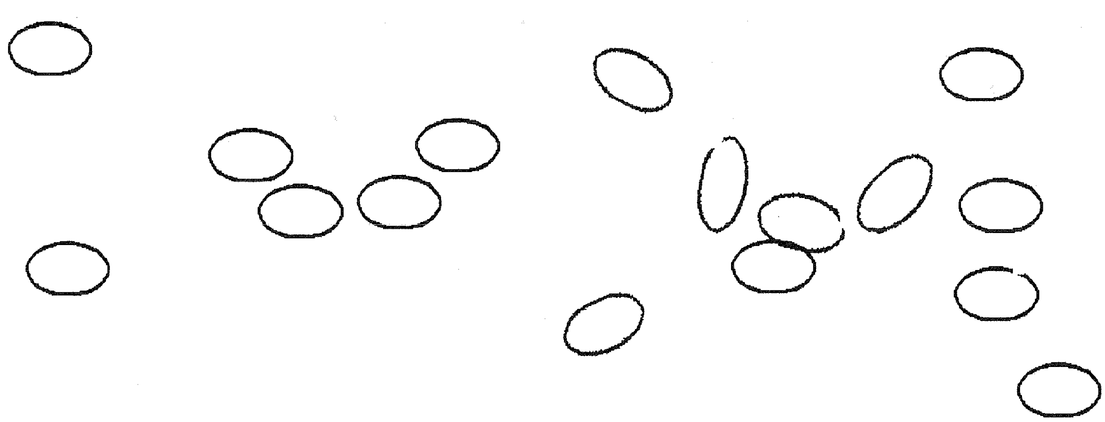

Figure 2: Ellipses recognized from Figure 1.

BIBLIOGRAPHY

Antonini, M., Barlaud, M., Mathieu, P., Daubechies, I., 1992. Image coding using wavelet transform. *IEEE Trans. On Image Processing*, 1, pp. 205–220.

Ballard, D., 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13, pp. 111–122.

Fletcher, L., Kasturi, R., 1988. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 10, pp. 910–918.

Gorman, J., Mitchell, R., Kuhl, F., 1988. Partial shape recognition using dynamic programming. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 10, pp. 257–266.

Grimson, W., Huttenlocher, D., 1990. On the sensitivity of the Hough transform for object recognition. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 12, pp. 255–274.

Huttenlocher, D., Klanderman, G., Rucklidge, W., 1993. Comparing images using the Hausdorff distance. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 15, pp. 850–863.

Jun, M., Xinru, L., Chengke, W., 1986. A new shape recognition system. In *Proceedings of the Eight International Conference on Pattern Recognition*, pp. 811–813.

Kasturi, R., Bow, S., El-Masri, W., Shah, J., Gattiker, J., Mokate, U., 1990. A system for interpretation of line drawings. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 12, pp. 978–991.

Kasturi, R., Shih, C., Fletcher, L., 1986. An approach for automatic recognition of graphics. In *Proceedings of the Eight International Conference on Pattern Recognition*, pp. 877–879.

Krzyzak, A., Dai, W., Suen, C. Y., 1990. Unconstrained Handwritten Character Classification Using Modified Back Propagation Model. In *Frontiers in Handwriting Recognition*, pp. 145-151, CENPARMI, Concordia University.

Le Cun, Y., Boser, B., Henderson, D., Howard, R. E., 1990. Constrained neural network for unconstrained handwritten digit recognition, In *Frontiers in Handwriting Recognition*, pp. 145-151, CENPARMI, Concordia University.

Levine, M., 1982. *Vision in Man and Machine*. McGraw-Hill, Inc., New York.

Lin, C., Chellappa, R., 1987. Classification of partial 2-D shapes using Fourier descriptors. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 5, pp. 686–690.

Okazaki, A., Kondo, T., Mori, K., Tsunekawa, S., Kawamoto, E., 1988. An automatic circuit diagram reader with loop-structured-based symbol recognition. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 10, pp. 331–341.

Rauber, T., 1994. Two-dimensional shape description. Technical Report GR UNINOVA-RT-10-94, Universidade Nova de Lisboa.

Reeves, A., Prokop, R., Andrews, S., Kuhl, F., 1988. Three-dimensional shape analysis using moments and Fourier descriptors. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 10, pp. 937–943.

Reiss, T., 1991. The revised fundamental theorem of moment invariants. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 13, pp. 830–834.

Rumelhart, D., McClelland, J., 1986. *Parallel distributed processing: explorations in the microstructure of cognition. Volume 1: Foundations*. MIT Press.

Said, F. N., Asselin, D., Suen, C. Y., 1995. Recognition of alphanumeric characters on engineering drawings. In *Fifth annual IRIS_PRECARN conference*, Vancouver, Canada, June 13-15.

Samet, H., Soffer, A., 1994. A legend-driven geographic symbol recognition system. In *Proc. of the 12th IAPR International Conference on Pattern Recognition*, pp. 350–355.

Shimotsuji, S., 1990. A robust drawing recognition system based on contour shape analysis. In *Proceedings of the Tenth International Conference on Pattern Recognition*, pp. 717–719.

Suen, C. Y., Legault, R., Nadal, C., Cheriet, M., Lam, L., 1993. Building a new generation of handwriting recognition systems, Pattern recognition letters, 14, 4, April.

Wu, R., Stark, H., 1986. Rotation and scale invariant recognition of images. In *Proceedings of the Eight International Conference on Pattern Recognition*, pp. 92–93.

Yu, Y.H., Samal, A., Seth, S., 1994. Isolating symbols from connection lines in a class of engineering drawings. *Pattern Recognition*, 27, pp. 391–404.