Intl. Journal of Document Analysis and Recognition. 1(2):89-101,June 1998

MAGELLAN: Map Acquisition of GEographic Labels by Legend ANalysis

Hanan Samet[*], **Aya Soffer**[**]

Computer Science Department and
Center for Automation Research and
Institute for Advanced Computer Science
University of Maryland at College Park
College Park, Maryland 20742
e-mail: {hjs,aya}@umiacs.umd.edu

Received: January 5, 1998 / Revised March 18, 1998

Summary. A system named MAGELLAN (denoting Map Acquisition of GEographic Labels by Legend ANalysis) is described that utilizes the symbolic knowledge found in the legend of the map to drive geographic symbol (or label) recognition. MAGELLAN first scans the geographic symbol layer(s) of the map. The legend of the map is located and segmented. The geographic symbols (i.e., labels) are identified, and their semantic meaning is attached. An initial training set library is constructed based on this information. The training set library is subsequently used to classify geographic symbols in input maps using statistical pattern recognition. User interaction is required at first to assist in constructing the training set library to account for variability in the symbols. The training set library is built dynamically by entering only instances that add information to it. MAGELLAN then proceeds to identify the geographic symbols in the input maps automatically. MAGELLAN can be fine-tuned by the user to suit specific needs. Recognition rates of over 93% were achieved in an experimental study on a large amount of data.

Key words: Map Recognition, Document Analysis, Object Recognition, Image Databases, Geographic Information Systems (GIS)

1 Introduction

The paper map has long been the traditional storage and retrieval medium for geographic data. Today, we are seeing the emergence of geographic information systems (GIS) as a replacement. One of the important issues in this field is how to integrate existing paper maps into a GIS. In particular, we would like to store scanned images of paper maps (termed *map images*) and to be able to retrieve portions of these maps based on the information that they convey, termed *retrieval by content*. An example query is "find all map images containing camping sites within 3 miles of fishing sites". We refer to such a system as a *map image information system* (*MIIS*). Such a system does not attempt to fully convert paper maps into one of the common GIS data formats (e.g. vector), and then let the GIS handle the queries and discard the map images. Such a conversion is very time-consuming and labor-intensive and is most likely not feasible for a very large collection of maps such as those stored in map libraries [13]. Instead, the goal is to retrieve map images based on content from a database that contains a large number of heterogeneous maps.

In order to support retrieval by content of map images, the maps must be interpreted to some degree when they are inserted into the database. In this paper, we describe a system called *MAGELLAN* (denoting *Map Acquisition of GEographic Labels by Legend ANalysis*) built by us that uses the legend of the map to drive the geographic symbol (or label) recognition for this purpose. MAGELLAN first locates the legend of the map and segments it. The geographic symbols are identified, and their semantic meaning is attached. An initial training set library is constructed based on this information. The training set library is subsequently used to classify geographic symbols in input maps using statistical pattern recognition [6]. User interaction is required at first to assist in constructing the training set library to account for variability in the symbols. Users may also change some of the classifier parameters at this stage to suit their specific application. Subsequently, MAGELLAN proceeds to automatically identify the geographic symbols in the input maps that use the same legend.

The emphasis of our approach is on utilizing the legend of the map to build a flexible and adaptive system that extracts symbolic information from map layers,

[*] The support of the National Science Foundation under Grant IRI-97-12715 is gratefully acknowledged.
[**] The support of the National Science Foundation under Grant CDA-950-3994 is gratefully acknowledged.
Correspondence to: Aya Soffer

rather than trying to vectorize the entire map and interpret every object found in it (some of these objects may not appear in the legend). The goal of MAGELLAN is to extract contextual cues from the map layer that can be used to index the composite maps in a map image information system. The input to MAGELLAN are raster images of the separate map layers. The output of MAGELLAN is a logical representation of a map image that that can be used by a map image information system to automatically index both the composite and layer images. See [22] for a description of a companion system called MARCO (denoting MAp Retrieval by COntent) that utilizes MAGELLAN and provides retrieval by content of map images.

In this paper we outline the general framework of MAGELLAN and describe its components. Although the methods that we use are not entirely new, the flexible and adaptive framework and the integration of these methods for map indexing are new and comprise our main contributions. One of the features that make MAGELLAN flexible and easily adaptive to different types of maps is its ability to generate more than one candidate classification for each symbol and most importantly to use these candidates in the retrieval process. Furthermore, users can fine-tune the parameters of the classifier such as the range of the search space, the maximum number of candidate classifications, and the minimum certainty of classification at an early stage in order to improve the recognition rate for their particular maps and for their specific application. We also developed an error classification paradigm which includes a new type of error termed an *addition error* to account for the fact that due to the assignment of more than one classification to an input symbol, it may be assigned a wrong classification in addition to being assigned a correct one. While having multiple classifications in the MIIS may improve the correct retrieval rates, addition errors will cause retrieval of extra results from the database, and thus it is important to account for these errors in the evaluation method. The usefulness of the flexibility of MAGELLAN for improving recognition was evaluated by conducting an experimental study. As part of this study, we make use of our error classification paradigm to measure the tradeoff in terms of the classification accuracy that results from different choices for the classifier parameters.

At this time, MAGELLAN can only handle point symbols. In particular, in our research we focused on symbol layers which contain geographic symbols that represent campsites, hotels, recreation areas, etc. In order to index by other layers that contain additional types of symbolic information such as roads, bodies of water, etc., other methods that are suitable for interpreting this kind of symbolic information need to be developed. MAGELLAN can be easily adapted to interpret other graphical documents and we have used similar methods for the interpretation of floor plans [21].

The rest of this paper is organized as follows. Section 2 lays out the background for this problem and discusses related work. Section 3 outlines the main components of MAGELLAN. Section 4 discusses the geographic symbol classification process. Section 5 presents our evaluation method along with experimental results. Section 6 contains concluding remarks.

2 Background and Related work

One of the most common means of data acquisition from paper maps is by use of a digitizing tablet. This process is very time-consuming, and labor-intensive. Optical scanners have also been put to use for this purpose. Once the maps have been scanned, the raster data is usually converted into vector format with very heavy user intervention in order to assure the quality of this conversion [25].

There has been much research in recent years on automating this acquisition process. Most researchers have focussed on raster-to-vector conversion [17]. Unfortunately, this does not always yield accurate and useful results. One of the reasons for these inaccurate results is that the conversion should be accomplished by some knowledge, which we term *map recognition*. One problem in map recognition is that a paper map is mostly an abstraction. The information found in maps is mainly symbolic rather than an accurate graphical description of the region covered by the map. The symbolism that is used in maps was designed for visual interpretation and does not always lend itself to blind machine interpretation. For example, the width of a line representing a road has little to do with the road's actual width. Instead, most often, the width of the road on the map is determined by the nature or type of the road (i.e., highway, freeway, rural road, etc.). The color and size of city names on the map convey information about the population of a city. As another example, graphic symbols are often used to indicate the location of various sites such as hospitals, post offices, recreation areas, scenic areas, etc. The actual key to interpreting this symbolic information is usually found on the map itself in the map's legend.

Another problem in map recognition is the high level of obstruction of geographic symbols due to the map-making process. A map is composed of several layers. The symbols in each layer in most cases do not occlude each other. However, once these layers are composed, the objects from the different layers may intersect and occlude each other thereby making the problem of segmentation and object recognition very complex. To alleviate this problem, we use a *layered approach* to map recognition. The input to MAGELLAN are raster images of the separate map layers. These layers can be obtained by a computational process or from the original map sources which were composed by an overlay-like process of separate layers to yield the composite map. The process of obtaining these separate map layers is not addressed by this paper. The extra step required to get the separate layers is well worth the effort as the results of map recognition on the map separates will most likely be an order of magnitude better than those that would result from using the composite map.

Although most prior research in map recognition has concentrated on skeletonization and vectorization methods [1,27], some research has been done on separating the layers of scanned maps [26]. The maps included road maps [10] and cadastral maps [4,11]. In the latter, the focus was on locating polygons representing parcels of land, buildings, and roads. In [28] an algorithm is presented to separate text strings of various fonts, sizes and styles from other graphic data in map images. A system for interpreting topographic maps is described in [7]. This system separates composite maps by colors, and processes each color separately. These color separates do not necessarily correspond to the original thematic layers thereby complicating the analysis of each layer. A system for automatic extraction of semantic information from land register maps is described in [15]. The system uses explicit knowledge about the map, which is derived from the legend of the map, map drawing rules, and knowledge about the functionality of objects in these maps. This system is specialized for land register maps. The user has to explicitly build the semantic model that is used to perform the classification of map objects.

In [12] a system that can be used to retrieve information from paper-based maps is described. The focus of this system is on query-driven map recognition. A portion of a scanned paper map is analyzed as a result of a query requesting information from the geographic area that corresponds to this portion of the map (e.g., city name). One part of this process may involve symbol recognition, such as identifying the symbol for a city. The process that is used to identify this symbol is not described in detail. It is most likely tailored towards a specific map and based on template matching. In contrast, our approach is legend-driven which makes it easily adaptive to any symbols that are found in the map legend.

In [29] a method for extracting geographic symbols from topographic maps based on multiangled parallelism (MAP) is presented. The MAP operation method performs parallel calculations on directional feature planes. Symbols are extracted using a reformalized parallel version of the Hough transform on these directional planes. This method is computationally-intensive. Running on a dedicated image processor, it takes 18 minutes to perform the first step of this process for a 640×512 image. Several more minutes are required for feature extraction for each symbol. While the results that are presented using this method seem robust and do not require separate layers, the time required to process each image makes this method difficult to use for a large collection of maps.

The above efforts are primarily academic. In addition, there are several companies that provide scanning services that include raster-to-vector conversion for scanned map data. SMARTSCAN, Scangraphics, and Neuralog are among the companies in this field. Unfortunately, the systems that these companies use are proprietary and the research involved in their development is unpublished. The products that these companies deliver (namely the map data in GIS format) are usually very accurate. These excellent results are attributed to a combination of manual editing of the maps beforehand, elaborate conversion algorithms, and complex error-checking logic with interactive human guidance. This procedure is expensive and not always appropriate for the purpose of indexing large heterogeneous sets of map images.

Before proceeding further, it is worthwhile to reemphasize that the focus of our approach is on utilizing the legend of the map to extract symbolic information from map layers, rather than trying to vectorize the entire map and interpret every object found in it. Our goal is to provide a flexible and adaptive system that can perform quick conversion of paper maps from a physical representation to a logical representation whose result can be used to index map images.

3 MAGELLAN System Overview

The input to MAGELLAN is a raster image of the symbols layer. This map image is divided into *tiles* of size 512×512 pixels (Figure 1). Note that it is possible for a symbol to lie in two tiles. In order to simplify our presentation, here we assume that this does not occur. If this is a concern, then an overlap of half the size of the largest symbol is required between tiles. These tiles are processed one-by-one. MAGELLAN (Figure 2) has two phases, the legend acquisition phase and the symbol classification phase, corresponding to the processing of legend and non-legend tiles, respectively.

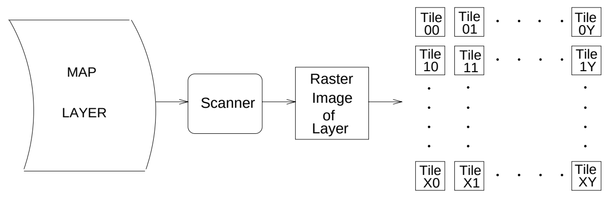

Fig. 1. Map layer acquisition and splitting process.

3.1 Legend Acquisition Phase

The purpose of the legend acquisition phase is two-fold. The first is for the user to indicate which symbols of the legend are of importance to the application. These symbols are termed *valid symbols*. Any other symbols that are found in map tiles but were not pointed out by the user at this stage are termed *invalid symbols*. The second purpose of the legend acquisition phase is to construct an initial training set library that is subsequently used in the symbol classification phase. This initial training set library contains a feature vector corresponding to one instance of each valid symbol along with its semantic meaning (also termed *classification*). This is the classification that MAGELLAN should output for each instance of this valid symbol that is subsequently input into it. For invalid symbols, a special classification termed *undefined* is used (i.e., invalid symbols should be classified as undefined by MAGELLAN). All other classifications (i.e., those that correspond to valid symbols) are termed *valid classifications*.

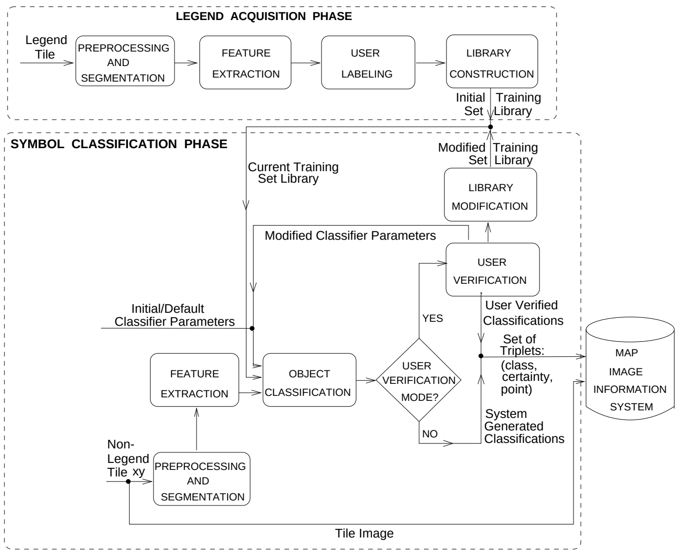
LEGEND ACQUISITION PHASE
Legend Tile
PREPROCESSING AND SEGMENTATION
FEATURE EXTRACTION
USER LABELING
LIBRARY CONSTRUCTION
Initial Training Set Library
SYMBOL CLASSIFICATION PHASE
Modified Training Set Library
Current Training Set Library
LIBRARY MODIFICATION
Modified Classifier Parameters
USER VERIFICATION
Initial/Default Classifier Parameters
YES
User Verified Classifications
Set of Triplets: (class, certainty, point)
MAP IMAGE INFORMATION SYSTEM
FEATURE EXTRACTION
OBJECT CLASSIFICATION
USER VERIFICATION MODE?
NO
System Generated Classifications
Non-Legend Tile xy
PREPROCESSING AND SEGMENTATION
Tile Image

Fig. 2. Block diagram of MAGELLAN (Map Acquisition of GEographic Labels by Legend ANalysis)

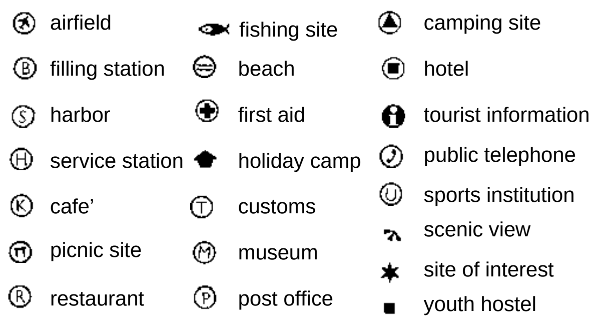

Fig. 3. Geographic symbols and their semantic meaning.

The legend acquisition phase is preceded by an operation that identifies the tile(s) containing the legend. These tile(s) serve as the input to the legend acquisition phase. Next, the legend tile(s) are segmented. A feature vector is computed for each connected component. The user identifies those symbols that are found in the legend that are of importance to the application. While a symbol may be composed of more than one connected component, we currently assume that the symbols may be uniquely distinguished from each other by just one of these connected components which serves as a representative of the symbol. The user identifies this representative connected component for each symbol. In order to select this representative symbol we make use of an interesting characteristic of symbols used in maps. In particular, many of these symbols are composed of a circle (or rectangle) enclosing one or more small shapes (e.g., the beach and hotel symbols in Figure 3 which shows the geographic symbols used in our test case along with their semantic meaning). One of the connected components that can be chosen as the representative of such a symbol is the interior of the circle with the small shapes considered as holes in this object (termed a *negative symbol*). For example, the "beach" symbol in Figure 3 would be represented by the interior of a circle with the two wiggly lines as holes. Currently we cannot distinguish between two symbols that differ only in the spatial configuration of the same components if they are not enclosed in a circle or rectangle. One optional solution for this is to artificially enclose such a symbol in a circle and use the negative symbol as the representative component. For more details on negative symbols, see [23].

Once the representative component is selected, the semantic meaning (*classification*) is attached to the feature vector corresponding to the connected component that represents each symbol. An initial training set library is constructed containing one instance of each valid symbol. The process of locating the legend and deriving the semantic meaning of the symbols in the legend is currently done manually. Since user verification and fine-tuning is a key component in maximizing the reliability of symbol classification, having the user assist in locating and analyzing the legend is not too onerous. However, we would like to automate parts of this process in the future so that it will become even simpler.

3.2 Symbol Classification Phase

Each non-legend tile is input into the symbol classification phase. This phase may operate in two modes. In the *user verification mode*, the user verifies the classifications before they are input to the map image information system (MIIS). The training set is modified to correct the erroneous classifications. The user may also fine-tune some of the classifier parameters such as the range of the search space, the maximum number of candidate classifications, the minimum certainty values, and the feature noise tolerance values while working in user verification mode. In the *automatic mode*, the classifications are generated automatically and input directly to the MIIS. The user determines the mode in which the phase operates. In general, the first tiles will be interpreted in user verification mode. Once the user is satisfied with the recognition rate achieved, the phase is placed in automatic mode to process the remaining tiles. There are five basic modules in this phase.

3.2.1 Preprocessing and Segmentation Module

In the preprocessing and segmentation module, various image processing techniques may be applied to enhance the image. These may include applying noise reduction filters, edge closing, thinning, etc. [19]. In our test case, we did not need to perform any of these operations since the images were rather clean. Next, the image is segmented into its constituent elements using a connected component labeling algorithm (e.g., [19]). The output of this module is a labeled image in which each pixel has a region number as its value. Regions that are smaller than a certain threshold are labeled 0.

3.2.2 Feature Extraction Module

The input to this module is the labeled image output by the preprocessing and segmentation module. For each region in the labeled image, a set of *features* is computed. MAGELLAN uses four global descriptors (first invariant moment, circularity, eccentricity, and rectangularity) and three local shape descriptors (horizontal gaps per total area, vertical gaps per total area, and ratio of hole area to total area). These features are commonly used to describe shapes [14], and we found them to be effective in discriminating between different geographic symbols [23]. These features are all invariant to scale and translation. In addition, most of them are also invariant to rotation. The results of the feature computation are composed into a *feature vector*. The center of gravity (i.e., centroid) of each region is also computed. The x and y coordinate values of this location are termed a *location vector*. The output of this stage is a *feature descriptor* that is composed of the feature vector which is a point in the n-dimensional *feature space*, and the location vector which is a point in the 2-dimensional *location space*.

3.2.3 Object Classification Module

The input to this module is the feature descriptor (feature and location vectors) output by the feature extraction module, the current training set library, and some classifier parameters. The training set library, constructed by the legend acquisition phase, initially consists of one feature vector for each geographic symbol along with its semantic meaning (i.e., its *classification*). This is the class that MAGELLAN should assign to each instance of the same symbol. Depending on the quality of the raster image, this may suffice. However, in most cases, the instances of each symbol vary and thus several instances of each symbol are required in order to build a representative training set library that will produce reasonable recognition rates.

The current training set library is used to assign candidate classifications to each input feature vector. A value approximating the certainty of the correctness of these classifications is attached to each classified object (see Section 4 for more details). The output of this module consists of the classifications that were made, the certainty of the classifications, and the corresponding location of the symbols on the map. Recall that in cases where a symbol is composed of more than one connected component, the user identifies one of these connected components as representative of the symbol. This component is used to classify the symbol in the classification phase. The remaining components should be classified as undefined since they do not have instances in the training set library. It is interesting to note that although two symbols may touch or even overlap, as long as their representative connected components do not touch or overlap the classification will work properly. For example, consider the symbols for “first aid” (circle with a cross inside) and “beach” (circle with the two wiggly lines) in Figure 3. Assume that the “first aid” symbol is represented by the cross component and the “beach” symbol is represented by its interior (negative symbol). The two symbols could potentially touch each other (i.e., the exteriors of the circles touch or intersect), while the representative connected components do not touch each other. However, if the representative connected components do touch or overlap, then the classification will fail as in this case the segmentation module will merge the two connected components into one. In the maps that we worked with, symbols in general did not touch. However, for maps where this occurs on a regular basis, other methods would be required to resolve this problem. One possible solution is to rely on more local features rather than global features.

3.2.4 User Verification Module

This module is only active when MAGELLAN is operating in user verification mode. The input to this module are the feature descriptors and the classifications resulting from the object classification module. An image containing the classification having the highest certainty value for each symbol in the input image is composed. This image is displayed next to the original input image. The user visually inspects these images and indicates which symbols have been classified incorrectly, and informs MAGELLAN of the proper classification for each such symbol. The user may view some additional information such as the certainty of the classification and whether there were several candidate classifications. The user may then change some of the classifier parameters. These include the search range, the maximum number of candidate classifications, the minimum certainty values, and the feature noise tolerance values (see Section 4 for a detailed description of these parameters). The user may then opt to reclassify the same tile using these new parameters in order to study the effect of their modifications. This module outputs the location, classification (as approved or corrected by the user), and a certainty of 1 for all the symbols found in the input map tile. This information is transfered to the map image information system. In addition, this module outputs those feature vectors corresponding to the symbols that MAGELLAN misclassified along with their correct classifications. This information is passed on to the library modification module. Finally, the current classifier parameters (whether they were modified or not) are output and passed to the classification module for use in its next invocation.

3.2.5 Library Modification Module

This module is only active when MAGELLAN is operating in user verification mode. The input to this module is a list of feature vectors along with their corresponding classification. These vectors are used to classify subsequent input symbols and thus they comprise a part of the training set for the classification process. Notice that only feature vectors of symbols that could not be classified correctly using the current training set library are added to the library. Feature vectors of correct classifications of a symbol are not added to the library as the library already has good representative feature vectors for these symbols (either from the legend acquisition phase or from prior invocations of the classification and library modification modules). This method of dynamically constructing the training set library ensures that the training set library will remain small without compromising the results of classifications using this training set library. The reason for this is that there is no redundant information in the training set library. Hence it is very efficient and yields results similar to those obtained using a condensing technique to minimize the size of the training set library [6]. The training set library is stored as an adaptive k-d tree [9,20]. Each feature vector (not the location) is stored as a point in this data structure. The output of this module is the current training set library, which is used by subsequent invocations of the object classification module.

3.3 Map image Information System (MIIS)

The input to the MIIS is a set of points, corresponding to the locations of the symbols found in the maps. For each point, its possible classifications and a certainty value approximating the correctness of each possible classification are given. This input comes either directly from the object classification module or from the user verification module depending on the mode in which MAGELLAN is operating. The MIIS uses this information to index the map images and to support retrieval by content of maps based on both contextual and spatial information (e.g., [24].

4 Classifying Geographic Symbols

Geographic symbols identified in the map tiles are classified using a modification of the *weighted several-nearest neighbor classifier* [3] termed a *weighted bounded several-nearest neighbor classifier.* This classifier makes use of two pre-defined constants: α, which is a neighborhood-size factor that determines the search range for neighbors based on the distance to the nearest neighbor, and β, which is a bound that determines the maximum distance allowed between the feature vector of an input symbol and its several-nearest neighbors in the training set library. This classifier first finds the feature vector F^{L_N} in the training set library (TSL) that is nearest to the feature vector of the input symbol F^I. Let D be the weighted Euclidean distance between F^{L_N} and F^I given by

$$D = dist(F^{L_N}, F^I) = \sqrt{\sum_{i=1}^{N} w_i (F_i^{L_N} - F_i^I)^2}.$$

where $F_i^{L_N}$ is the ith feature of the training set library vector F^{L_N}, F_i^I is the ith feature of the input vector F^I, and w_i is the weighting factor of ith feature of the feature vector. The weighing factor is computed so that features with a smaller variance have a larger weight as described in [5]. Next, the classifier finds the set φ_I^L of all training set library feature vectors F^L whose distance to F^I is less than the smaller of α times D, and β. Formally:

$$\varphi_I^L = \{F^L \mid F^L \in TSL \wedge dist(F^L, F^I) < min(\alpha \times D, \beta)\}.$$

The range defined by $min(\alpha \times D, \beta)$ is termed the *$\alpha\beta$-neighborhood.* The $\alpha\beta$-neighborhood is a fuzzy search bound that is determined by the distribution in feature space of the feature vectors in the training set. If the distance between an input symbol and its nearest neighbor is small, then the classifier will only search for additional neighbors in a relatively small range since it already has a good candidate. On the other hand, if the the distance between an input symbol and its nearest neighbor is large, then the classifier will search for additional neighbors in a relatively large range in order to find more candidates. Since invalid symbols (not found in the legend) need to be classified as undefined, we must specify an absolute bound on this search space. β is this bound, and any symbol that does not have any neighbors closer than β are classified as undefined.

Each feature vector $F^L \in \varphi_I^L$ (the set of symbols in the $\alpha\beta$-neighborhood) is given a vote, whose strength is inversely proportional to its distance from the feature vector of the input symbol F^I, given by

$$Vote_{F^L} = \frac{1}{dist(F^L, F^I)}.$$

The votes of all feature vectors that belong to the same classification C_i are summed giving:

$$Votes_{C_i} = \sum_{F^L \in \varphi_I^L \wedge class(F^L) = C_i} Vote_{F^L}.$$

If $\varphi_I^L = \emptyset$ (i.e., the distance to the nearest neighbor was $> \beta$), then the input symbol is classified as undefined. A certainty value between 0 and 1 is computed for each candidate classification C_i found in the $\alpha\beta$-neighborhood of the input feature vector. This value approximates the certainty that the input vector belongs to C_i. The certainty value is calculated by normalizing the value of $Votes_{C_i}$ with respect to some minimal and maximal acceptable values of $Votes_C$ for any of the possible candidate classifications C. The maximal acceptable vote value is determined by selecting a minimal required distance d_{min}, for a "sure" classification (i.e., if $dist(F^L, F^I) < d_{min}$, then F^I will be assigned the training set library classification corresponding to F^L with certainty 1). Hence, the maximal value for $Votes_C$ is $1/d_{min}$. The minimal acceptable vote value is determined by selecting a maximal allowed distance d_{max} for a classification to be considered as a candidate (i.e., if $dist(F^L, F^I) > d_{max}$, then the training set library classification corresponding to F^L will not be considered as a candidate classification for F^I at all). Hence, the minimal value for $Votes_C$ is $1/d_{max}$. d_{min} and d_{max} are thus in effect noise tolerance levels in feature space and are application-specific. Two features that are closer than d_{min} are considered the same (i.e. we assume that the difference is due to noise). On the other hand, two features that are further than d_{max} cannot represent the same classification. The motivation for calculating the certainty values in this manner is that the certainty values must rank the candidate classifications with respect to one another not only in one invocation of the classifier, but must do so with respect to the candidate classifications of all other invocations of the classifier as well. Therefore, some global method of normalizing certainty values is required.

The nearest neighbors of the training set library (stored as a bucket variant of an adaptive k-d tree[9, 20]) that are within the $\alpha\beta$-neighborhood are found using a modification of the priority k-d tree search algorithm [2]. This algorithm visits the buckets of the k-d tree in increasing order of their distance from the input feature vector. The search is complete when the distance from the input feature vector to the closest remaining bucket is outside of the range determined by α, and β. The classification module outputs all of the candidate classifications along with their certainty. The classification with the highest certainty value is considered the best classification for the input vector using the weighted bounded several-nearest neighbor classifier.

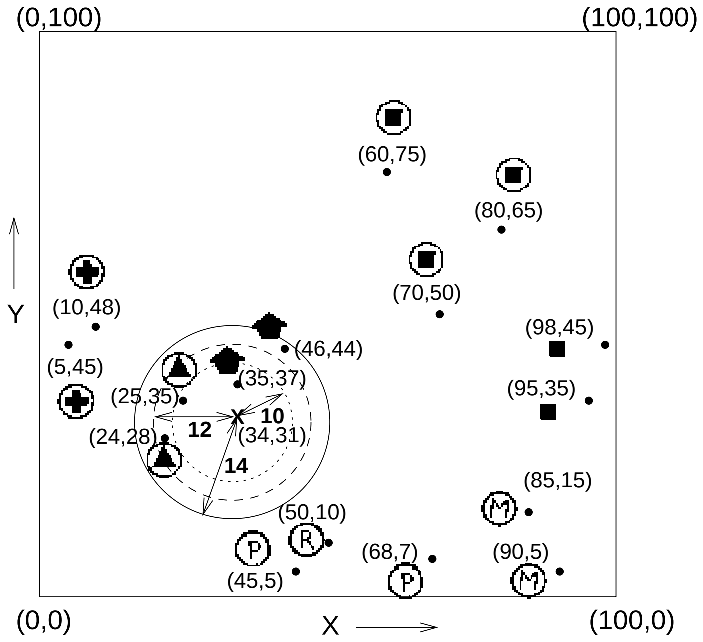

Fig. 4. Example of classifying an input symbol X using a weighted bounded several-nearest neighbor classifier. $\alpha = 2$, $\beta_1 = 10$, $\beta_2 = 14$. The dotted circle is the $\alpha\beta_1$-neighborhood, the dashed circle is the $\alpha\beta_2$-neighborhood, and the solid circle is the range defined by β_2.

Figure 4 demonstrates the classification process using an example training set of symbol instances in 2-space (i.e., we are assuming a feature vector of just 2 features). Let $X = (34, 31)$ be the feature vector (in 2-space) of an input symbol. Let $\alpha = 2$, $\beta = 10$, and $w_1, w_2 = 1$ (i.e., both features are assigned an equal weight in the distance computation). The classifier assigns candidate classifications to X as follows. First, the feature vector F^{L_N} in the training set library (TSL) that is nearest to the feature vector of X (F^I) is found. In this case, $F^{L_N} = (35, 37), D = \sqrt{37} = 6.082$. The classifier next finds the set φ_I^L of all library feature vectors F^L whose distance to F^I is less than the smaller of $\alpha \times D$, and β (i.e., in the $\alpha\beta$-neighborhood). In this case, $sizeof(\alpha\beta$-neighborhood$)$ = $min(12.16, 10) = 10$. Thus, $\varphi_I^L = \{(35,37),(25,35)\}$, where (35,37) is an instance of the symbol "arrow" (holiday camp) and (25,35) is an instance of the symbol "triangle" (camping site). $Votes_{arrow} = 1/\sqrt{37} = 0.164$, $Votes_{triangle} = 1/\sqrt{97} = 0.101$. Thus, X will be assigned classification "arrow" with a higher certainty than classification "triangle". Recall that the certainty value is calculated by normalizing the value of $Votes_{C_i}$ with respect to some minimal and maximal acceptable values of $Votes_C$ for any of the possible candidate classifications C. These values are determined according to d_{min} and d_{max}, the minimal and maximal acceptable values for $dist(F^L, F^I)$. Thus, to calculate the certainty values for this example we must assign some values to these parameters. Assuming that we allow a noise tolerance of 2 units in each feature in the 2D feature space, d_{min} = $\sqrt{8}$ and thus $max(Votes_C) = 1/\sqrt{8} = 0.353$. Assuming a maximum allowable difference of 10 units in each feature in the 2D feature space, $d_{max} = \sqrt{200} = 14.142$ and thus $min(Votes_C) = 1/\sqrt{200} = 0.071$. Therefore we get, $certainty(X \in arrow) = \frac{0.164-0.071}{0.353-0.071} = 0.33$ and $certainty(X \in triangle) = \frac{0.101-0.071}{0.353-0.071} = 0.106$ since we normalized 0.164 and 0.101, respectively.

However, $\beta = 14$, then $sizeof(\alpha\beta$-neighborhood$)$ = $min(12.16, 14) = 12.16$. Point (24,28) which is another instance of "triangle" will now also be included in φ_I^L. Thus, $Votes_{arrow} = 1/\sqrt{37} = 0.164$, $Votes_{triangle} = 1/\sqrt{97} + 1/\sqrt{109} = 0.198$. In this case, X will be assigned classification triangle with a higher certainty than classification arrow. Assuming the same d_{min} and d_{max}, we get $certainty(X \in arrow) = 0.33$ and $certainty(X \in triangle) = 0.45$. Notice that by summing the votes when there are several neighbors from the same class, $Votes_{C_i}$ may potentially exceed $max(Votes_C)$. In practice this only happens if there are many instances in the training set that are very close to the input symbol, since the contributions of instances that are far from the input symbol are very small. In this case, we set the certainty of this classification to 1 since there is very strong evidence that it is in fact the correct one.

5 Experimental Study

MAGELLAN was tested on the red sign layer of the GT3 map of Finland. The scale of the map is 1:200,000. The layer was scanned at 240dpi. Although this resolution may not be fine enough for general topographic maps [7], it was sufficient for our case since the smallest valid symbol in the map layer was 0.03 inch (corresponding to 8

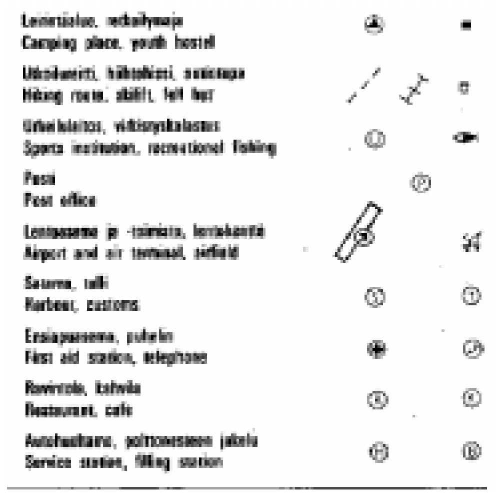
Leirintäalue, retkeilymaja
Camping place, youth hostel
Ulkoilureitti, hiihtolatu, autiotupa
Hiking route, skilift, left hut
Urheilulaitos, virkistyskalastus
Sports institution, recreational fishing
Posti
Post office
Lentoasema ja -terminaali, lentokenttä
Airport and air terminal, airfield
Sauna, tulli
Harbour, customs
Ensiapuasema, puhelin
First aid station, telephone
Ravintola, kahvila
Restaurant, cafe
Huoltoasema, polttoaineen jakelu
Service station, filling station

Fig. 5. Legend portion containing tourist symbols

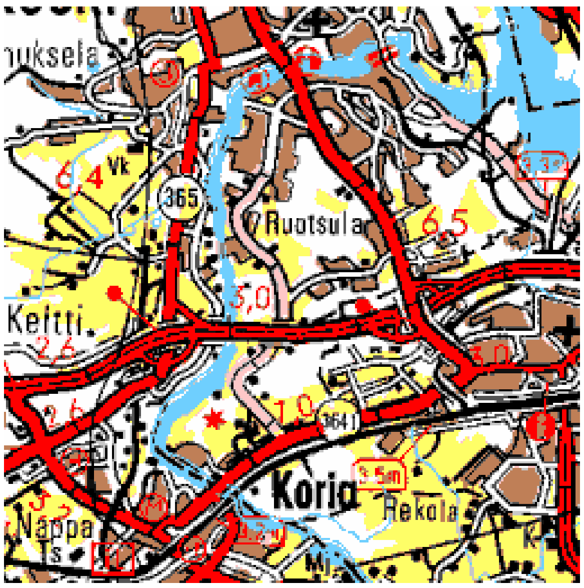

Fig. 6. Example map tile - all layers

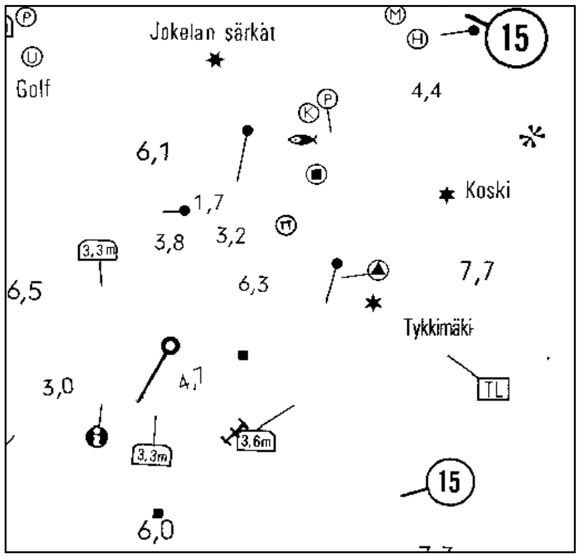

Fig. 7. Example map tile - red sign layer

pixels at 240dpi). Figure 5 is a portion of the map's legend relevant to the sign layer (note that this is only part of the legend, the full legend contained the 22 symbols in Figure 3). Figure 6 is a sample tile while Figure 7 shows the extracted red sign layer. Notice that there are many symbols in the map tile that are not found in the legend. These are mainly numbers, names, and markers that are related to other layers. These symbols, termed *invalid symbols*, should all be classified by MAGELLAN as undefined as explained in Section 3.1. The layer was split into 425 tiles of size 512×512. These tiles were examined in a random order to give MAGELLAN a chance to see a large variety of symbols while operating in user verification mode as some symbols tend to appear clustered in the map. The legend was identified manually, and the classifications were attached to the feature vectors representing the geographic symbols that MAGELLAN should identify. There were 22 such symbols. MAGELLAN processed the first 60 tiles in user verification mode. At that stage, the training set contained 100 instances of symbols and the current recognition rate was deemed adequate. The remaining tiles were processed automatically. See Section 5.3 for the results of this fully automatic recognition. MAGELLAN was implemented using Khoros [18], an integrated software development environment for information processing and visualization. The image processing and classification were executed on a Sparc 10 running UNIX. The average execution time for the main steps for each 512×512 tile were as follows: connected component labeling – 12 seconds, feature extraction – 4 seconds, and classification – 0.2 seconds.

The results of this classification were input into MARCO [22] (denoting MAp Retrieval by COntent). It is a map image database that provides retrieval of map images according to their contents by means of spatial and non-spatial queries. For example, the user may request to display all tiles containing camping sites within 3 miles of fishing sites.

5.1 Evaluation Method

In order to evaluate MAGELLAN, the following three error categories, common in optical character recognition (OCR) [16], were defined: *substitution errors* — a valid input symbol was assigned an incorrect valid classification (e.g., picnic site instead of post office); *deletion errors* — a valid input symbol was classified as undefined; *insertion errors* — an invalid input symbol was assigned one of the valid classifications rather than being classified as undefined.

Recall that MAGELLAN outputs all of the candidate classifications of the feature vector corresponding to an input symbol that were found in its $\alpha\beta$-neighborhood (see Section 4). As part of our experiment, we were interested in determining whether inserting into the MIIS more than one candidate classification per symbol with an appropriate certainty value yields better results than just inserting the class with the highest certainty value. In this case, we insert the same point more than once with different certainty values and classifications. If any of these candidate classifications is the correct one, then this is not counted as a substitution error since the correct classification is in the MIIS, although not with the highest certainty value. In order to account for the fact that we now have additional points with erroneous classifications in the MIIS, we define an additional error category – *addition errors* — an input symbol (can be either valid or invalid) was assigned more than one valid classification. Each additional classification (after the first one) is counted as one addition error.

Substitution errors and deletion errors may cause the MIIS to overlook tiles that should be retrieved for a given query. Insertion errors and addition errors may cause the MIIS to retrieve superfluous map tiles for a given query. In the context of an MIIS, the impact of insertion and addition errors is not as severe as that of substitution and deletion errors. Recall that the purpose of map recognition is to enable the MIIS to retrieve just those map portions that are relevant to a given query. Thus, retrieving too many tiles is not as harmful as missing tiles. The user can always weed out those tiles that do not actually conform to the query.

5.2 Experiment Description

50 sample tiles were chosen from the tiles that were not used for training MAGELLAN. These tiles were input into MAGELLAN. For each symbol in each tile, MAGELLAN output all of the candidate classifications of the neighbors in the $\alpha\beta$-neighborhood of the symbol's feature vector along with a certainty value as described in Section 4. In the first stage of the experiment, only the candidate classification with the highest certainty value was compared to the correct classification as found in the raw image. The number of errors of each type for each tile was recorded. In the next stage, all of the candidate classifications were considered. The number of errors of each type was recorded for two cases. In the first case, only the best and second-best (highest and second-highest certainty values) classifications were considered. In the second case, all classifications were considered. The certainty values assigned by MAGELLAN for each correct classification and incorrect classification were also noted.

This experiment was repeated five times varying the value of β, which corresponds to the maximum distance in the normalized (having unit width) feature space allowed between the feature vector of an input symbol and its neighbors in the training set (termed the *search bound*). Any symbol whose nearest neighbor is not within this search bound is classified as undefined. The values selected for β (the search bound) were 0.02, 0.05, 0.1, 0.2, 0.4. These values are relative to a search space having unit width. As the search bound increases, more neighbors representing more classifications will be found in the range. Therefore, we expect to have fewer substitution and deletion errors, at the cost of more insertion and addition errors. The value of α (the neighborhood size factor) was set to 2 throughout all of the experiments. In other words, for each experiment, the classifier considers all of the feature vectors in the training set library whose weighted Euclidean distance from the feature vector of the input symbol is less than the smaller of 2 times the distance to the input vector's nearest neighbor, and the particular value of β for that experiment. The value 2 was chosen empirically as a reasonable value for α. Varying this value will most likely change the experimental results but not the general trend.

5.3 Results of Experiment

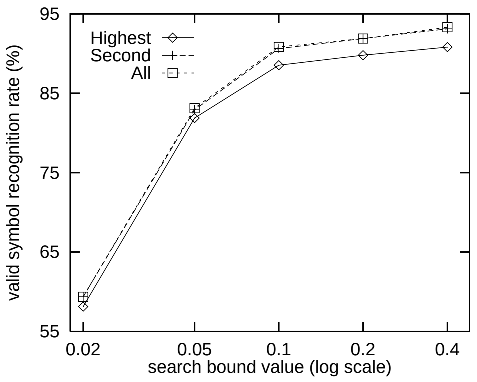

Fig. 8. Valid symbol recognition rate for various search bound values.

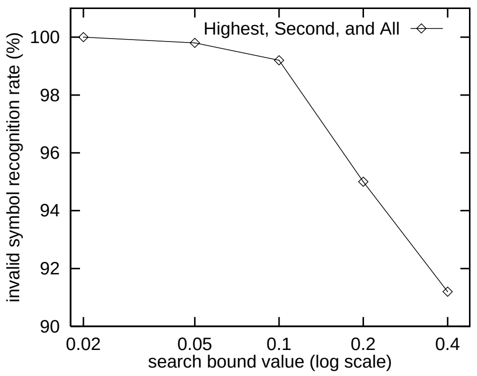

Fig. 9. Invalid symbol recognition rate for various search bound values.

Figures 8 and 9 show the recognition rate for valid and invalid symbols, respectively. Recall that valid symbols are those that the user indicated as important to the application in the legend acquisition phase. Any other symbols that are found in map tiles but are not instances of symbols that were pointed out by the user at that stage are invalid symbols. The percentages reported here are of the total number of valid and invalid input symbols in the 50 test images. The valid symbol recognition rate indicates what percent of the valid input symbols were assigned the correct classification. The invalid symbol recognition rate indicates what percent of the invalid input symbols were in fact classified by MAGELLAN as undefined. The "Highest" plot shows the recognition rate when considering only the classification with the highest certainty value. The "Second" plot shows the rate when considering the classifications with the highest and second-highest certainty value. The "All" plot shows the rate when considering the classifications of all of the neighbors in the $\alpha\beta$-neighborhood regardless of their certainty value.

As expected, as the search bound value increases, the valid symbol recognition rate increases and the invalid symbol recognition rate decreases. The reason for this is that there are potentially more candidate classifications within the $\alpha\beta$-neighborhood when the search bound is larger. Therefore, the chance that a feature vector corresponding to a symbol from the correct classification for a valid input symbol lies in the neighborhood increases. Similarly, the chance that a feature vector corresponding to some valid symbol from the library will lie in the neighborhood of an invalid input symbol also increases. This results in the invalid symbol being assigned a valid classification rather than undefined thereby decreasing the invalid symbol recognition rate.

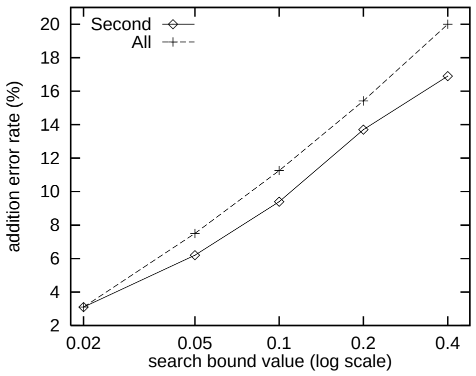

Fig. 10. Percent of the classifications output by MAGELLAN that are attributed to multiple classifications for the same symbol (i.e., an addition error occurred) for various search bound values.

From Figure 8 we also see that considering more than one candidate classification increases the valid symbol recognition rate. Notice however that the difference between the valid symbol recognition rate when considering just the first two candidate classification and the valid symbol recognition rate when considering all of the candidate classifications is very small. This means that in almost all of the cases that the best classification was incorrect, and there was more than one candidate classification, the second-best classification was the correct one.

Although considering more than one candidate classification improves the valid symbol recognition rate, it also has a negative side effect. In particular, it introduces addition errors since more than one classification may be recorded in the MIIS per symbol. Figure 10 shows the percent of the total number of classifications that were output by MAGELLAN that are attributed to multiple classifications for the same input symbol (i.e., an addition error occurred) for various search bound values. For small search bound values, this number is small. However, it grows significantly for larger search bound values, with slightly larger values when considering all candidate classifications rather than just the first two candidate classifications.

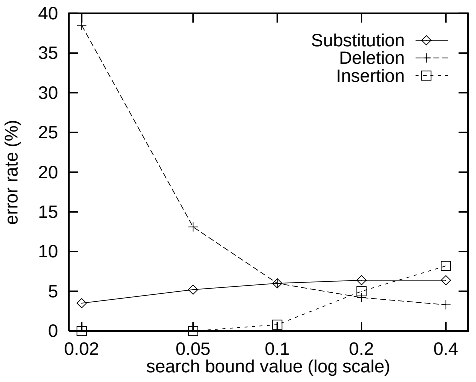

Fig. 11. Error rates when considering only the classification with the highest certainty value for various search bound values.

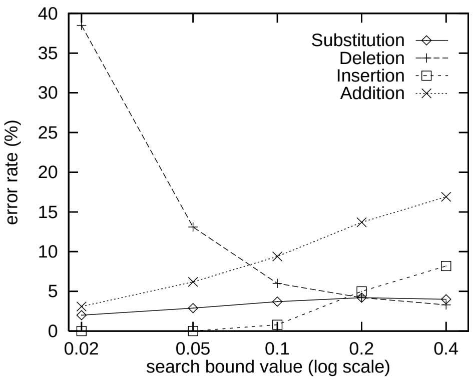

Fig. 12. Error rates when considering the highest and second-highest classifications for various search bound values.

Figures 11, 12, and 13 enable us to analyze the cause of the erroneous valid symbol classifications. These figures show the rate of the various error types as a function of the search bound value when considering the classification with the highest certainty value, the classifications with the highest and second-highest certainty values, and all classifications, respectively. Observe that the substitution error rate is only slightly effected by the search bound value, whereas the deletion error rate is highly effected by it. From this observation we may conclude that the increase in the valid symbol recognition rate with an increase in the search bound value is mainly at-

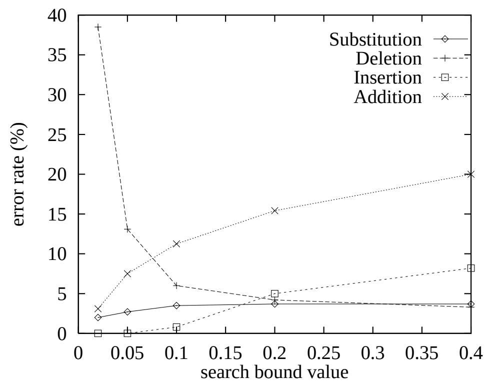

Fig. 13. Error rates when considering all candidate classifications for various search bound values.

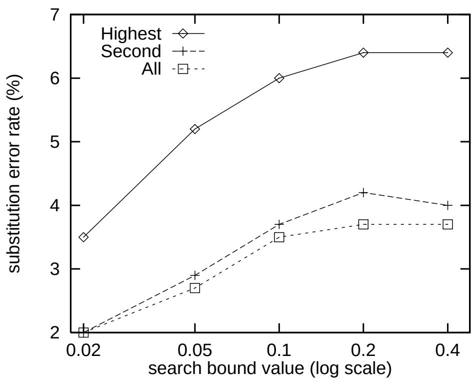

Fig. 14. Percent of input valid-symbols assigned the w valid classification (i.e., a substitution error occurred) foi ious search bound values.

tributed to a sharp reduction in the number of dele errors instead of a significant change in the numb substitution errors. The only way to decrease the su tution error rate is to consider more than one of the didate classifications found among the neighbors in $\alpha\beta$-neighborhood as can be seen in in Figure 14 whic ports the substitution error rate for various search bo values when considering the highest ranked, first and ond highest ranked, and all candidate classification

The search bound value of 0.1 appears to be bes this data set. A valid symbol recognition rate of 91% an invalid symbol recognition rate of 99% were achi with this window size (assuming the classifications the highest and second-highest certainty values in $\alpha\beta$-neighborhood were taken into account). In addi 10% of the symbols that were assigned valid classi tions were results of multiple classifications for a symbol (i.e., an addition error). Clearly, the ideal se bound value for our test data set may not be the best for a different data set. Furthermore, as our experim indicate, the particular choice of search bound value have profound effects on the various error rates. Th fore, MAGELLAN's capability of letting users set value of α and β (which determine the search range) ing user verification enables users to fine-tune MAG LAN for their particular data set and choice of featu as well as to their intended application. If it is cri for the MIIS not to miss any tiles when responding to a query, then a larger search bound value should be selected. This will cause the MIIS to overlook fewer tiles at the cost of retrieving superfluous tiles that will need to be weeded out manually. If accuracy is not as important as reducing the time required to weed out the tiles manually, then a smaller search bound value should be selected. Note that in the experiments that we report in this paper, we only varied β. It is, however, quite clear that the particular choice of α will also effect the various error rates. Thus, both α and β can be set by the user.

From our results, it is apparent that for our test data only the candidate classifications with the highest and second-highest certainty values should be considered (i.e., transferred to the MIIS). The improvement in the valid symbol recognition rate when considering all candidate classifications rather than just the first two was very small, while the addition error rate became larger. Thus, considering all candidate classifications does not seem to be beneficial. This conclusion may not be valid for other data sets (as with the particular choice of β), and thus the maximum number of candidate classifications may also be set while working in user verification mode.

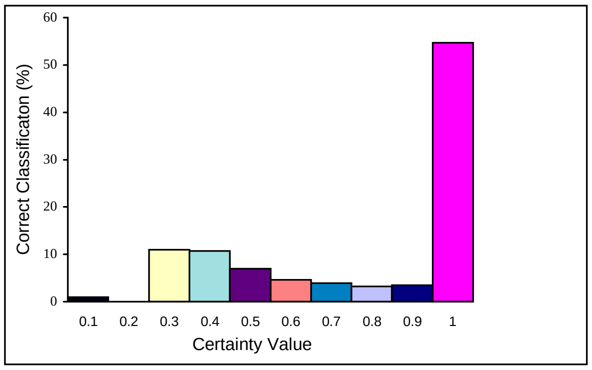

Fig. 15. Correct valid symbol classifications per certainty value range.

Figure 15 shows the certainty values given to valid input symbols that were assigned the correct classification (i.e., no error occurred). Similarly, Figure 16 shows the certainty values given to symbols when an insertion or substitution error occurred. The majority (more than 50%) of correct classifications were given a certainty value above 0.9, whereas the certainty values of the erroneous classifications are concentrated at the lower end of the certainty value range. The minimum certainty value required in order to pass a candidate classification to the MIIS is another parameter that users can set. For the data set that we tested, selecting a minimum certainty value of 0.3 would result in eliminating many of the substitution and insertion errors, thereby automatically weeding out most of the superfluous tiles while overlooking only a small number of the required tiles.

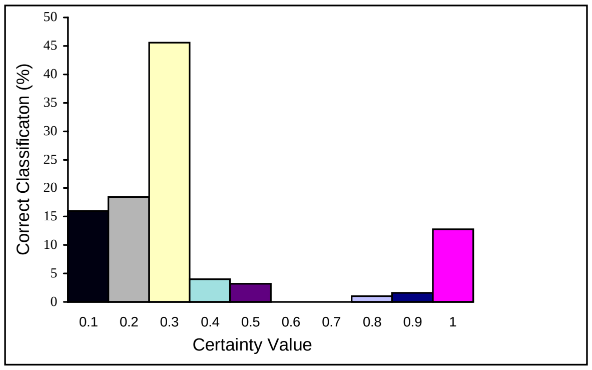

Fig. 16. Erroneous classifications per certainty value range.

Recall that the certainty values are computed based on a predetermined noise tolerance value (d_{min} and d_{max}) for the particular features that are used. Since this tolerance may also vary between applications and data sets, users can also set these values.

It is important to note that the results that we report here are for one particular training set and for the particular features that we have used. We experimented with various training sets, and the results reported here were consistent with the results that we encountered with other training sets. In order to achieve lower error rates, more features that better characterize the symbols would be required. This would result in more computationally intensive image preprocessing and classification steps. The insertion error rate may be reduced by removing some of the invalid symbols from the maps before the classification phase. Since in our case, most of these symbols are strings of letters or numbers, it may be possible to separate them from the images using methods for separating text from graphics such as those described in [8, 28]. In our case, since the emphasis was on enabling quick processing of a large number of images, we did not employ such methods. It is possible to get recognition rates of 100% by opting to always run MAGELLAN in user verification mode. In this case, the user would need to check every tile before it is inserted into the database. Although this was not the intended use of MAGELLAN, it can be used in this way effectively.

6 Concluding Remarks

A system called MAGELLAN (denoting Map Acquisition of GEographic Labels by Legend ANalysis) has been described. MAGELLAN utilizes the fact that most of the data found in maps is symbolic and that the key to interpreting the symbols can be found in the legend of the map. MAGELLAN is efficient and flexible. Users may fine-tune the performance of MAGELLAN to their requirements by setting the search bound parameters (α and β), the minimum certainty values, the maximum number of candidate classifications, and the feature noise tolerance values to fit their particular data sets and application. An experimental study was conducted on a large amount of data in order to study the effects of varying these parameters and to evaluate the performance of MAGELLAN . The experimental results showed that once the parameters are fine-tuned, MAGELLAN can achieve recognition rates of 93% with little user intervention. However, with more user intervention it is possible for MAGELLAN to reach 100% recognition rates.

Although it may seem that simple template matching [14] may have been sufficient for our application, this is not the case. Different instances of the same symbol may vary in scale and orientation. Therefore, we have chosen to use statistical pattern recognition with features that are invariant to scale and many features that are also invariant to rotation. MAGELLAN can be easily adapted to interpret other graphical documents and we have used similar methods for the interpretation of floor plans [21]. MAGELLAN is designed for map layers containing geographic symbols. In order to provide full map recognition, other methods need to be developed to interpret layers containing additional types of symbolic information such as roads, bodies of water, etc. Once this is done, the results can be integrated into a MIIS in order to provide map indexing based on additional geographic information.

Acknowledgements.

We are grateful to Karttakeskus, Map Center, Helsinki, Finland for providing us the map data and to the helpful comments of the referees.

References

1. S. V. Ablameyko, B. S. Beregov, and A. N. Kryuchkov. Computer-aided cartographical system for map digitizing. In *Proceedings of the Second International Conference on Document Analysis and Recognition.*, pages 115–118, Tsukuba Science City, Japan, October 1993.
2. S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 573–582, Arlington, VA, January 1994.
3. J.L. Blue, G.T. Candela, P.J. Grother, R. Chellappa, and C.L. Wilson. Evaluation of pattern classifiers for fingerprints and OCR applications. *Pattern Recognition*, 27(4):485–501, April 1994.
4. L. Boatto, V. Consorti, M. Del Buono, S. DiZenzo, V. Eramo, A. Esposito, F. Melcarne, M. Meucci, A. Morelli, M. Mosciatti, S. Searci, and M. Tucci. An interpretation system for land register maps. *Computer*, 25(7):25–33, July 1992.
5. G.L Cash and M. Hatamian. Optical character recognition by the method of moments. *Computer Vision, Graphics, and Image Processing*, 39(3):291–310, September 1987.
6. P. Devijver and J. Kittler. *Statistical Pattern Recognition.* Prentice-Hall, Englewood-Cliffs, NJ, 1982.

7. N. Ebi, B. Lauterbach, and W. Anheier. An image analysis system for automatic data acquisition from colored scanned maps. *Machine Vision and Applications*, 7(3):148–164, 1994.
8. L. A. Fletcher and R. Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):910–918, November 1988.
9. J.H. Friedman, J.L. Bentley, and R.A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226, September 1977.
10. M. Ilg. Knowledge-based interpretation of road maps. In *Proceedings of the Fourth International Symposium on Spatial Data Handling*, pages 25–34, Zurich, Switzerland, July 1990.
11. R. D. T. Janssen, R. P. W. Din, and A. M. Vossepoel. Evaluation method for an automatic map interpretation system for cadastral maps. In *Proceedings of the Second International Conference on Document Analysis and Recognition*, pages 125–128, Tsukuba Science City, Japan, October 1993.
12. R. Kasturi and J. Alemany. Information extraction from images of paper-based maps. *IEEE Transactions on Software Engineering*, 14(5):671–675, May 1988.
13. M. L. Larsgaard. *Map Librarianship: an Introduction.* Libraries Unlimited, Littleton, CO, 2nd edition, 1987.
14. M.D. Levine. *Vision in Man and Machine.* McGraw-Hill, New York, 1982.
15. G. Maderlechner and H. Mayer. Automatic acquisition of geographic information from scanned maps for GIS using frames and semantic networks. In *Proceedings of the 12th International Conference on Pattern Recognition*, volume II, pages 361–363, Jerusalem, Israel, October 1994.
16. S. Mori, C. Y. Suen, and K. Yamamoto. Historical review of OCR research and development. *Proceedings of the IEEE*, 80(7):1029, July 1992.
17. D. J. Peuquet. An examination of techniques for reformatting cartographic data part 1: The raster-to-vector process. *Cartographica*, 18(1):34–48, January 1981.
18. J. Rasure and C. Williams. An integrated visual language and software development environment. *Journal of Visual Languages and Computing*, 2(3):217–246, September 1991.
19. A. Rosenfeld and A.C. Kak. *Digital Picture Processing.* Academic Press, New York, second edition, 1982.
20. H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, Reading, MA, 1990.
21. H. Samet and A. Soffer. Automatic interpretation of floor plans using spatial indexing. In S. Impedovo, editor, *Progress in Image Analysis and Processing III*, pages 233–240. World Scientific, Singapore, 1994.
22. H. Samet and A. Soffer. MARCO: MAp Retrieval by COntent. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):783–798, August 1996.
23. A. Soffer and H. Samet. Negative shape features for image databases consisting of geographic symbols. In C. Arcelli, L. P. Cordella, and G. Sanniti di Baja, editors, *Advances in Visual Form Processing*, pages 569–581, Singapore, 1997. World Scientific.
24. A. Soffer and H. Samet. Pictorial query specification for browsing through image databasess. In *Proceedings of the Second International Conference on Visual Information Systems*, pages 117–124, San Diego, California, December 1997.
25. J. Star and J. Estes. *Geographic Information Systems*, chapter 6, pages 85–91. Prentice-Hall, Englewood Cliffs, NJ, 1990.
26. S. Suzuki and T. Yamada. MARIS: Map recognition input system. *Pattern Recognition*, 23(8):919–933, August 1990.
27. N. Tanaka, T. Kamimura, and J. Tsukumo. Development of a map vectorization method involving a shape reforming process. In *Proceedings of the Second International Conference on Document Analysis and Recognition*, pages 680–683, Tsukuba Science City, Japan, October 1993.
28. T. Xu and X. Lin. A new algorithm separating string from map images. In *Proceedings of the Second International Conference on Document Analysis and Recognition*, pages 910–913, Tsukuba Science City, Japan, October 1993.
29. H. Yamada, K. Yamamoto, and K. Hosokawa. Directional mathematical morphology and reformalized hough transformation for analysis of topographic maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):380–387, April 1993.

Hanan Samet received the B.S. degree in engineering from the University of California, Los Angeles, and the M.S. Degree in operations research and the M.S. and Ph.D. degrees in computer science from Stanford University, Stanford, CA. He is a Fellow of the ACM, the IEEE, and the IAPR (International Association for Pattern Recognition).

In 1975 he joined the Computer Science Department at the University of Maryland, College Park, where he is now a Professor. He is a member of the Computer Vision Laboratory of the Center for Automation Research and also has an appointment in the University of Maryland Institute for Advanced Computer Studies.

His research interests are data structures, computer graphics, geographic information systems, computer vision, robotics, programming languages, artificial intelligence, and database management systems. He is the author of the books *The Design and Analysis of Spatial Data Structures*, and *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*, both published by Addison-Wesley, 1990.

Aya Soffer is a research associate at the Center for Automation research at the University of Maryland College Park where she conducts research on image databases. She received the B.S degree in computer science from the Hebrew University of Jerusalem in 1986, and the M.S and Ph.D degrees in computer science from the University of Maryland at College Park in 1992 and 1995, respectively. During the period 1995-1997 she was a Research Assistant Professor at the University of Maryland Baltimore County as well as a Research Scientist in the Center for Excellence in Space Data Information Systems at the NASA Goddard Space Flight Center

She has developed a map image database system which supports retrieval by content of map images using a novel pictorial query specification tool. Her research interests include pictorial information systems, document analysis and recognition, digital libraries, geographic information systems (GIS), and non-traditional database systems.