COMBINING SOURCES OF EVIDENCE TO RESOLVE AMBIGUITIES IN TOPONYM RECOGNITION IN CARTOGRAPHIC MAPS*

Alexander GELBUKH *‡ SangYong HAN‡
* Natural Language Processing Lab, Center for Computing
Research CIC–IPN, UPALM Zacatenco,
Mexico City, 07738 MEXICO
E-mail: gelbukh@cic.ipn.mx; www.gelbukh.com
and
‡Department of Computer Science and Engineering, Chung-Ang University, 221 Huksuk-Dong, DongJak-Ku,
Seoul, 156-756, KOREA
E-mail: hansy@cau.ac.kr

Serguei LEVACHKINE †
† Image Processing and PR Lab, Centre for Computing
Research–IPN, UPALM Zacatenco, Ed. CIC,
Mexico City, 07738 MEXICO
E-mail: palych@cic.ipn.mx

ABSTRACT

Graphical documents such as cartographic maps contain a great variety of textual elements appearing in different spatial positions, in different fonts, sizes, and colors, touching and overlapping graphical symbols. This greatly complicates automatic optical recognition of such textual elements in the process of raster-to-vector conversion of graphical documents. In this work, we propose a method that combines OCR-based text recognition in raster-scanned maps with heuristics specially adapted for cartographic data to resolve the recognition ambiguities using various sources of evidence. Our goal is to form in the vector thematic layers geographically meaningful words correctly attached to the cartographic objects..

KEY WORDS

Raster-to-Vector Conversion, Geographic Information Systems, Optical Character Recognition, Spelling Correction

1. INTRODUCTION

Huge amount of geographic information collected in the last centuries is available in the form of maps printed or drawn on paper. To store, search, distribute, and view these maps in the electronic form they are to be converted in one of digital formats developed for this purpose. The simplest way of such conversion is scanning the paper map to obtain an image (a picture) stored in any of the raster graphical formats such as TIFF, GIF, etc. After that, a raster-to-vector conversion can be applied to include obtained vector maps into a Geographic Information System (GIS).

Though raster representation has important advantages in comparison with the hard copy form, it still does not allow semantic processing of the information shown in the map, for example:

- Search for objects: *Where is Pittsburgh? What large river is there in Brazil?*
- Answering questions on the spatial relations: *Are Himalayas in China? Is Nepal in Himalayas? Is a part of Himalayas in China?*
- Generation of specialized maps: *Generate a map of railroads and highways of France.*
- Scaling and zooming: *Generate a 1:125 000 map of Colombia. Now, show more details at the point under cursor.*
- Compression: *Objects such as points, arcs, or areas can be stored much more efficiently than pixels*

Note that these are semantic tasks rather than image manipulation. E.g., when zooming in or out, objects and, most importantly, their names should appear or disappear rather than become smaller or larger. Indeed, when zooming out the area of London, the name Greenwich should not become small to unreadable but should disappear (and appear in an appropriate font size when zooming in).

This suggests storing and handling of a map as a database of objects (points, arcs, areas, alphanumeric, etc.)—vector database—having certain properties, such as size, geographic coordinates, topology, and name. Specifically, the name of the object is to be stored as a letter string rather than a set of pixels as originally scanned from the hard copy. Thus, such vector representation can solve the listed above semantic tasks, but only to some extent [1].

* Work done under partial support of Mexican Government (CONACYT, SNI, IPN), Korean Government (KIPA Professorship for Visiting Faculty Positions in Korea), and ITRI of the Chung-Ang University. The first author is currently on Sabbatical leave at Chung-Ang University.

However, automatic recognition of such strings (toponyms) in the raster image of the map presents some particular difficulties as compared with the optical character recognition (OCR) task applied to standard texts such as books:

- The strings are out of context, which prevents from using standard spelling correction techniques based on the linguistic properties of coherent text. Often such strings are even not words of a (modern) language, which further limits applicability of the standard linguistic-based spelling correction methods [12].
- The background of the string in the map is very noisy since it can contain elements of geographic notation such as shading or hatching, cartographic objects such as cities or rivers, and even parts of other strings, e.g., name of a city inside of the area covered by the name of the country; see Figure 1.
- In addition, the letters of the string are not properly aligned but instead are printed under different angles and along an arc; this happens with the names of linear and area objects, e.g., rivers or countries; see Figure 1.
- Unlike standard task, in toponym recognition it is not only required to recognize the string itself but also to associate it with a specific cartographic object, e.g., city, river, desert, etc.

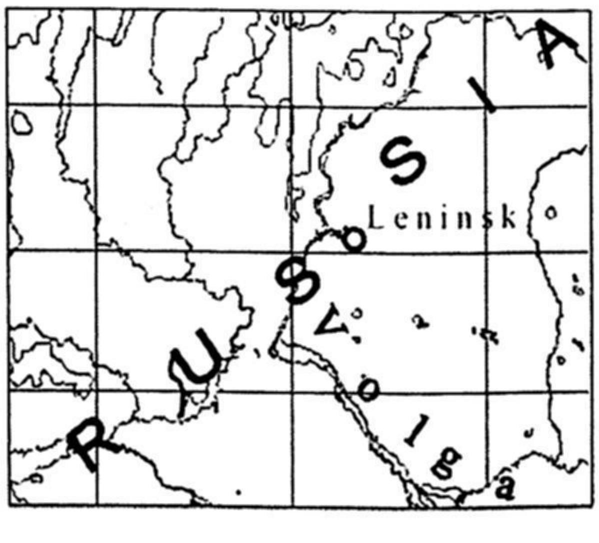

Figure 1. Intersection of string in a map

On the other hand, in many cases additional information is available that can give useful cues for ambiguity resolution. One of such information sources is existing databases (usually available from the country Government, postal service, etc.) providing spatial relationships between entities (e.g., a list of cities classified by administrative units) or even exact coordinates; see [14] for extensive discussion of this topic.

In this paper we discuss how such additional information can be used to workaround the problems arising in recognition of the inscriptions in the maps, associating them with specific cartographic objects, and importing information on these objects from available databases.

First we describe the general scheme of our method. Then we discuss various sources of evidence taking into consideration, when available, in error detection and correction: information of the existing names and linguistic information, on the distribution of the letters of the string in the source raster image, and pre-existing geographic information such as coordinates of objects. Then global verification of consistency of the recognition results is described. Finally, conclusions are drawn and future work directions are outlined.

2. PREVIOUS WORK AND PRESENT PAPER OVERVIEW

The text segmentation and its subsequent recognition in raster images are very difficult problems due to the presence of the text embedded in graphic components and the text touching graphics [2]. These challenging problems have received numerous contributions from the graphic recognition community [3]. However, there have not been yet developed any efficient programs to solve the task automatically. Thus, in the most works human operator is involved. For example, [4] proposes that the operator draws a line through the text, marking it as text and revealing its orientation.

In [5] and [6], the algorithms are developed to extract text strings from text/graphics images. However, both methods assume that the text does not touch or overlap with graphics. For maps, the problem is much more complex, since the touching or overlapping as well as many other character configurations are commonly presented in maps. That is why [7], [8], and [9] developed the methods for text/graphics separation in raster-scanned (color) cartographic maps.

In [9] a specific method of detecting and extracting characters that are touching graphics in raster-scanned color maps is proposed. It is based on observation that the constituent strokes of characters are usually short segments in comparison with those of graphics. It combines line continuation with the feature line width to decompose and reconstruct segments underlying the region of intersection. Experimental results showed that proposed method slightly improved the percentage of correctly detected text as well as the accuracy of character recognition with OCR.

In [7] and [8], the map is first segmented to extract all text strings including those that are touched other symbols and strokes. Then, OCR using Artificial Neural Networks (ANN) is applied to output the coordinates, size, and orientation of alphanumeric character strings present in the map. Then, four straight lines or a number of curves computed in function of primarily recognized by ANN characters are extrapolated to separate those symbols that are attached. Finally, the separated characters are input into ANN again for their final identification. Experimental results showed 95–97% of successfully recognized alphanumeric symbols in raster-scanned color maps.

In the present work, we use the output obtained with this method in combination with pre-existing geographical information in semantic analysis of ambiguities for "geographically meaningful" word formation. We focus on text processing rather than image processing.

The proposed system is based both on the traditional techniques used in the general-purpose OCR programs and on the techniques we developed especially for cartographic maps. In particular, Section 5 deals with the problems and solutions common to any OCR task. However, even in these cases there are some differences with respect to the usual OCR situation. The algorithm described in Section 5.1 (check against a dictionary of existing words) in our case has to deal with much more noisy strings than usual OCR programs developed for clean black-on-white running text. The same can be said of Section 5.2 (non-uniform spatial letter distribution): in maps the letters are often placed at significant distances one from another, cf. Figure 1; as well of Section 5.3 (check against the general laws of a given language): maps have many foreign or indigenous words that do not conform to the main language of the given territory.

In contrast, Section 4 is specific for maps. In Section 4.3 (check against geographic information such as expected coordinates) the consistency with the available information about the location of an object is used, which is specific for cartographic maps. Also the information on the expected type of the object (river, mountain, etc.) is used. In Section 4.4 (global consistency check) it is verified that each object is recognized only once. These techniques do not have direct analogs in standard OCR research and thus are contributions of our paper.

Finally, we do not use many techniques standard for usual text OCR, which are applicable to running text but not to toponyms in maps, for example: morphological and syntactic analysis, semantic consistency verification [13]; paragraph layout determination, etc. In a way, the new techniques we introduce in the Section 4 play the same role of verification of contextual consistency, but in the manner very specific to cartographic maps.

3. MAIN PROCEDURE

We rely on a basic OCR procedure[1] (not discussed here; see [1], [7], and [8]) that recognizes in the map individual letters and groups together the letters of a similar font and color located next to each other, thus forming a hypothetical string. In this process, errors of various types can be introduced; our purpose is to detect and correct them.

The recognition algorithm for the whole map works iteratively. At each step, the basic OCR procedure selects for processing the longest and most clearly recognized string and returns it for error correction and subsequent adding to the database being constructed. Upon its processing, the string is removed from the raster image, and the next string is selected. The algorithm stops when no more letter strings can be found in the raster image.

This design allows for recognition of the names of large areas, which are usually represented by large letters scattered across the corresponding area, with many names of smaller objects between the letters of the area name. In the example shown in Figure 1, first the word Leninsk will be recognized and removed from the image, then the word Volga, and only then the letters of the word Russia can be grouped together in a string.

The basic OCR procedure returns, for each string it recognizes, the string itself, e.g., "RUSSIA," and the geographic coordinates in the map of the frame containing each individual letter, e.g., R, U, etc.

After this process, two major issues arise:

- How to associate the textual objects found in the map with the geographical objects found in the same map? In Figure 1, what are the type (city, river, mountain, etc.) and the coordinates of the object called *Leninsk*? What is the name of the city located near the center of the map?
- How to detect and correct possible recognition errors in the textual elements?

To solve these problems, various sources of evidence are to be taken into account. In the following, we will consider each such source of evidence, first for the association problem and then for the error detection and correction problem.

4. ASSOCIATION OF A NAME WITH AN OBJECT

As we have assumed, the basic OCR procedure returns two types of information:

- *Geographical objects.* These can be of three types: punctual, linear, and area objects. For them, the basic OCR procedure returns the corresponding coordinates in the map (in pixels or in the corresponding geographical units) as will be discussed in the following.
- *Textual information.* The basic OCR procedure returns a string along with the coordinates (again, in

[1] Our method does not depend on how text strings have been extracted and recognized. Neither does it depend much on the type of graphical document being processed. It can be adapted to different subject domains.

pixels or in geographical units) of a box containing each of its letters.

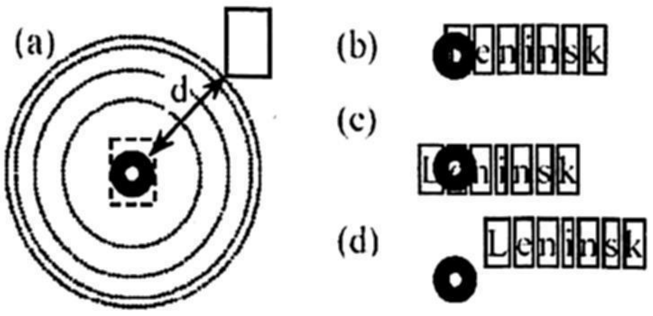

Figure 2. Simplified model for punctual objects

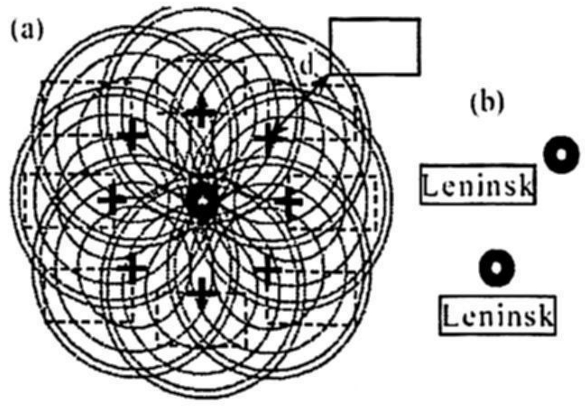

Figure 3. Improved model for punctual objects

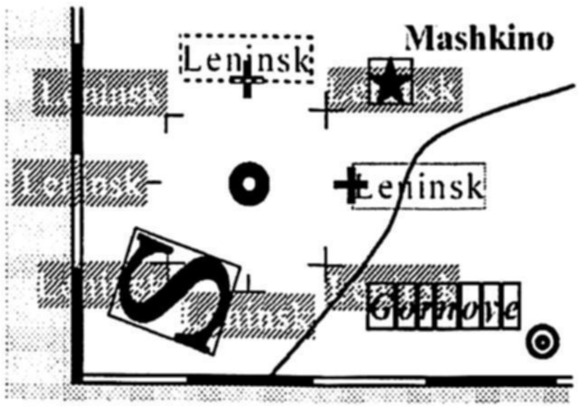

Figure 4. Constrained placement strategy

The next task for the map recognition system is to relate the strings (toponyms) with the objects found in the map. This is a non-trivial task due to several peculiarities.

First, it is highly heuristic since one needs to model the way in which the human cartographer assigned the labels to the objects. Second, not all objects have a corresponding label as well as not all labels correspond to objects detectable in the map.

The assignment procedure consists of two major parts: estimating of probability of a string to be related to an object, and final assignment of the strings to objects in a way that maximizes such probability. In the subsequent sections we shall consider these two tasks separately.

4.1 LIKELIHOOD OF RELATEDNESS BETWEEN A STRING AND AN OBJECT

Given a geographic object and a string, both along with their coordinates (in pixels or in geographical units), we can estimate the probability of that the string is related to the object. Using the Bayes formula, we can do it by modeling the process of placement of the names in the map by the cartographer.

Indeed, denote by R the event that the string is related to the object and by P the event that the cartographer placed the string to a specific position in the map (where we observed it). By Bayes formula, the desired probability is:

$$P(R \mid P) = P(P \mid R)\frac{P(R)}{P(P)} \qquad (1)$$

Since P(P) does not depend on a specific object and thus does not affect the disambiguation decisions, the desired probability is determined by the following two factors:

- P(P | R) reflects the strategy used by the cartographer to place the names of the objects next to the objects,
- P(R) reflects the relatedness of the name with the given object.

These values can be estimated heuristically taking into consideration various sources of evidence. We define these sources as mean proportional value:

$$P = \sqrt[n]{\prod_{i=0}^{k} P_i}, \qquad (2)$$

where P is P(P | R) or P(R), correspondingly, and Pi are the probabilities contributed according to each source of evidence.

In what follows, we discuss various independent sources of evidence used in our method.

4.2 SPATIAL EVIDENCE

To define the probability P = P(P | R) of placing the object's name in the observed specific position where it has supposedly been found in the map, we should model the strategy used by the cartographer for placing the name of this object. Then, we can assume that various (independent) random factors may cause the cartographer to deviate from the "optimal" position. The effect of various independent factors is approximated well by the normal distribution:

$$P = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{d^2}{2\sigma^2}}, \qquad (3)$$

where d is the distance from the actual inscription to the optimal position predicted by the model, and σ is a coefficient (dispersion) depending on the scale of the map and the fonts used; its selection is discussed in the following. If the model predicts several possible placements $x_1, ..., x_n$ with probabilities $p_1, ..., p_n$ and dispersions $\sigma_1, ..., \sigma_n$, then we assume:

$$P = \sum_{i=1}^{n} p_i \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{d_i^2}{2\sigma_i^2}}, \qquad (4)$$

where d_i are the distances from the observed placement to the corresponding coordinates x_i.

The placement strategies are different for punctual, linear, and are objects.

Punctual objects. For a punctual object (such as a city), which is represented by only one coordinate pair p, our previous work [15] suggested the following strategy of placing its name. We assumed that the inscription is expected to be next to the point p. Thus, we computed the distance d as minimum distance from the point p to any of the frames containing the individual letters of the string, as shown in Figure 2 (a). Though this simple model is a reasonably good approximation, it is not very precise. For example, both placements shown in Figure 2 (b) and (c) are predicted by the model to be optimal while (d) is not; this is contra-intuitive.

The model can be improved as shown in Figure 3 (a). First, we observe that the names of punctual objects (unlike the names of linear or area objects; see Figure 1) are aligned along a straight line; thus, instead of the frames of individual letters as in Figure 2, the frame containing the whole string can be considered. Second, we consider eight possible placement strategies shown in Figure 3 (a). The name is placed next to the object, not overlapping with the object, in some small distance from the object. This distance is approximately the size of one letter. We suggest that the dispersion σ_i from (4) should be also approximately of the size of a letter frame.[1]

With this improvement, examples of (locally) optimal placements are shown in Figure 2 (d) and Figure 3 (b), while the placements shown in Figure 2 (b) and (c) are, in accordance to our intuition, not optimal.

The eight strategies have different probabilities pi in (4). For example, in the languages with left-to-right writing system, the placements to the right of the object are preferred to those to the left. The procedure for determining these parameters is described here in Section 4.5

What is more, not all of the eight placement strategies can be possible in a specific environment. A specific placement strategy is not possible if the string would significantly overlap with any of other objects found in the map:

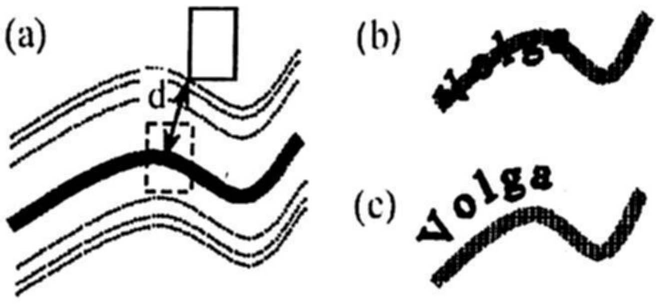

Figure 5. Simplified model for linear objects

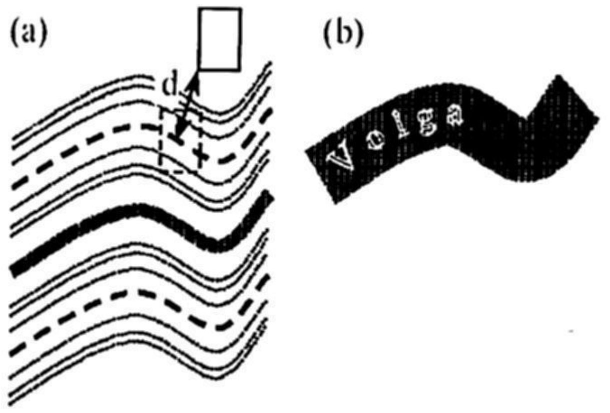

Figure 4. Improved model for linear objects

- *Letters of other textual elements.* The string may be placed between the letters of a string of a larger font, given that it does not overlap with individual letters.
- *Other punctual objects.* The string may, however, overlap with other linear or area objects.
- *Borders of the map.* The string cannot trespass beyond the area of the map.

An example is shown in Figure 4, where only two placement strategies are possible. In such cases, in the formula (4) the probabilities p_i of the impossible cases are set to zero and the other p_i are re-normalized. Alternatively, instead of setting the corresponding probabilities to zero, they can be significantly decreased (penalized).

Note that we assume that the operation of relating the names with the objects is performed after independent recognition of all objects and all strings in the map, so that the positions of other strings and objects are known at this moment.

Linear objects. For linear objects (such as rivers) represented by a sequence of coordinate pairs p_i, we suggest similar improvements over the procedure proposed in [15]. In the latter work, we indicated that the

[1] The model can be further improved taking into consideration that each individual distribution in (4) is not symmetrical: deviations that do not change the distance from the object are most probable, and toward the object are less probable that those away from the object. This can be done by a suitable deformation of the coordinate system; we omit here the details of this procedure.

slightly simplified way, to measure the distance between a letter and the broken line, two adjacent points x_i, x_{i+1} nearest to the letter are found and the distance from the letter to the straight line connecting the two points has been determined.

Similarly to the case of punctual objects, this would lead to a situation shown in Figure 5 (a), which incorrectly predicts the case (b) rather than (c) to be optimal. As in the case of punctual objects, we also suggest considering two placement strategies shown in Figure 6 (a), which correctly predict the case (c) and not (b) in Figure 5 to be optimal. The parameters (such as p_i and σ) and constraints (such as those shown in Figure 4) are treated much in the same way as in the case of punctual objects discussed above; we skip here the details.

An exception is the linear objects with the width significantly greater than the size of the letters in the string. In this case, the old model should be used, as shown in Figure 6 (b); namely, the string is expected to be found in the middle of the line. Note that our processing of such objects is different from that of area objects in that the string does not need to cover the whole length of the object.

Area objects. For an area object S (such as a province) represented by a sequence of coordinate pairs xi corresponding to its contour, our previous work [15] suggested the following approach. The inscription is expected to be in the middle of the area and the letters are expected to be distributed by the whole area. Thus, we can take $d = \iint_{S'} f(x,y)dxdy$ in (3), where *f(x,y)* is the minimum distance from the point *(x,y)* to any of the letters of the string. The integral is taken over the intersection S' of the area S and the whole area of the given map (in case a part of S proves to be out of the boundaries of the given map). Note that a similar integral along the contour would not give the desired effect. Since the area objects are much less numerous than other types of the objects in the map, we do not consider computational efficiency a major issue for our purposes. Neither precision is important for us. Thus, it is enough to compute the integral by, say, Monte-Carlo method.

Now we can re-interpret this procedure along the lines of the approach described in detail for the punctual and linear objects. Namely, the string that minimizes the integral above is the predicted "optimal" placement, with individual letters uniformly covering the surface of the area object. The observed placement can differ from the predicted one with the probability given by (3).

However, in this case we deal with a set of objects—individual letters—and not with one object, the whole string. Their distribution by the map can be considered independent. Thus, the probability of a specific configuration of n letters is:

$$P=\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{d_i^2}{2\sigma^2}}, \qquad (5)$$

where d_i are the distances between the predicted and actual location of individual letters.

This new interpretation allows for a meaningful choice of the parameter σ in (5), which in [15] was left undefined. The deviation in the placement of each letter can be of the order of about 1/3 of the distance between the letters in the predicted string, for the inscription not too look too misplaced.

Computationally, the task of finding of the optimal (predicted) string that minimizes the integral discussed above can be treated, for example, with 2*n*-dimensional gradient descent.

As in the case of linear objects, exceptions are to be considered. If the area object is too small in comparison with the font used in the string, it should be considered as punctual object. If the area object is similar to a line (very much longer in one dimension than in the other one), it can be treated as a thick or thin linear object, see Figure 6 (b).

4.3 APPROPRIATENESS OF A NAME FOR AN OBJECT

The previous section dealt with the component P(P | R) of (1) reflecting the placement strategy used by the cartographer. In contrast, this and the following sections deal with the component P = P(R), which reflects the appropriateness of a particular name for a particular object, independently of the physical location of the string on the map.

Each of the following subsections discusses a specific contribution P = P_i in the total probability. These contributions are combined by (2). In all cases, except for the next subsection, such probabilities are, though, binary: the combinations of a name and an object are classified into possible and impossible ones.

Typographic Evidence. As we have mentioned, the basic OCR procedure returns the coordinates of each letter. This can give us two characteristics of the recognized string:

- Whether the letters are aligned along a straight line,
- The distance between each adjacent pair of letters.

Only the names of linear and area objects (e.g., rivers or lakes), but not punctual objects (e.g., cities), can have non-linear letter alignment. Non-linear alignment is admitted for non-punctual objects but not required.

It is the responsibility of the basic OCR procedure to evaluate the probability P of that a string is linearly aligned, which is to be used in case of a punctual object. Note that this condition is not applicable to linear and area objects.

Notational Evidence. Notation in the map gives additional information to filter out impossible combinations of names and objects. In some maps, rivers are explicitly marked as "*river*" or "*r.*" and similarly mountains, peninsulas, etc. Specific font family, size, and color are usually associated with various types of objects (e.g., cities, and rivers). Though this information can provide very good filtering, it is not standard and is to be automatically learnt or manually specified for each individual map, which limits the usefulness of such filtering capability in a practical application.

Automatic learning of notation is discussed in Section 4.5. Alternatively, the system can provide the operator with the means to specify such notational elements, at least the prefixes such as "*river*". Similarly, font features for a specific type of objects can be automatically learnt from a large map or specified by the operator.

The importance of recognition of such notational information is two-fold. First, it helps filtering out impossible combinations: for example, the name of a punctual object cannot be specified as the name of a river. Another use of notational information is discussed in the next subsection.

Some precautions should be taken with such type of filters. For example, in Spanish rivers are marked as "*río*" 'river'; however the string *RÍO DE JANEIRO* should not be filtered out as a possible name of the city (given that capital letters are not properly distinguished in the map).

Geographic Evidence. This is a very powerful source of evidence, though it relies on extensive databases not always available. Suppose the string is found in a dictionary (database) that provides at least two types of spatial information on the corresponding object:

- Its inclusion in a larger area, such as a province, state, etc. These areas form a hierarchy.
- Its geographic coordinates.

This information can be used to verify that the object in question recognized in the map satisfied the constraints specified by the database for the string in question.

Note that when only the hierarchical information is available (for example, "*Jalapa city is in Oaxaca state*"), this can be used to filter out undesirable variants only if the coordinates are available for one of larger areas, one or more steps up the hierarchy (but small enough to serve for disambiguation). Alternatively, it might happen that the corresponding larger area has been earlier recognized in the same map. Unfortunately, due to the order of recognition from smaller to larger objects (see the beginning of Section 3), it is hardly probable. The corresponding check can be performed at the post-processing stage—global verification, see Section 4.4, when all areas have been already recognized.

In the best case, the full coordinate information is available in the dictionary for the object. Then the task of verification is greatly simplified, provided that the coordinate grid is reliably recognized for the given map.

The dictionary frequently contains several objects with same name, of the same or different type. When analyzing a map of *Canada*, the object corresponding to a recognized string *London* is to be a small Canadian city and not the large British city, so that the correct number of inhabitants for the object could be imported from the dictionary to the database being constructed. When analyzing an inscription *Moscow* in the coordinates (57°N, 35°E), its interpretation as a river rather than city is more probable.

Note that for correct identification of geographic information associated with a toponym, some information about notational conventions is important for addressing the dictionary. Indeed, for the string "*river Thames*" what is to be looked up in the dictionary is "*Thames*" and not "*river Thames*".

Linguistic Evidence. This is a substitute for the lack of knowledge on notation in a specific map. In some languages, the names of rivers, mountains, cities, etc., tend to follow some patterns that can be specified in the linguistic module of the recognition system. For example, in English a name ending in –town is more probable for a city than for a river. In Russian, a name ending in –ka is probable for a river or village, but not for a mountain. In Korean, a name ending in –do would probably indicate an island and –gan a river.

Obviously, these clues should be taken into account as factors in the total probability and not as rigid constraints (unless they are rigid constraints in the language in hand).

4.4 VERIFICATION OF GLOBAL CONSTRAINTS

After all inscriptions in the map have been recognized, some global constraints should be checked.

Uniqueness. To each object only one inscription should correspond. If two inscriptions have been associated with the same object, one or both of them is to be re-assigned. Even though the information on the probability of each of the two candidates is available at this point and could allow for automatic selection of one of the candidates, we believe that such conflicts should not be arbitrated automatically but the human intervention is to be requested instead. Of course, the probability information can be used to suggest most likely variant to the human operator.

An exception from this rule is linear objects such as long rivers. Several inscriptions can be assigned to such an object if their text is the same, the distance between them is much larger than their lengths, and their length are much smaller than the length of the object (river).

Inclusion. The hierarchical information available from the dictionary (see Section 4.3) can be applied at this point. Recall that our algorithm recognizes the names of, say, cities before recognition of the names of areas. So at the time of recognition of the string "*Xalapa*" the information "*Xalapa City is in Veracruz State*" could not be checked since we did not know yet where *Veracruz* State is in the map. Now that all strings have been recognized, this information can be checked (we already know where *Veracruz* is) and the error discussed in Section 5 (*Xalapa* mistaken for *Jalapa* recognized in Oaxaca State) can be detected.

4.5 MODEL CALIBRATION AND AUTOMATIC LEARNING OF PARAMETERS

The process described in the previous sections depends on a number of parameters, such as dispersion values or notational conventions. For their automatic learning iterative model calibration is used.

First, some approximate values are set as discussed in the previous sections. Then the automatic procedure of recognition of the map is executed. As a result, a (possibly incorrectly) recognized map is obtained.

Our hypothesis is that many of the elements in such a map will be recognized correctly from the first attempt. So statistics built for the results of this recognition—such as the average deviation of the strings from the predicted locations—is expected to be a good approximation of the real values.

With this new information, the parameters of the model (such as dispersion values) are adjusted, and the automatic recognition is performed again. The process is repeated iteratively a predefined number of times or until convergence. Since the results of the whole procedure are discrete values—associations between strings and objects—convergence can be indicated by repetition of exactly the same result.

In our previous work, we have successfully applied this procedure to learning the parameters of a syntactic parser for natural language sentences [10].

With this procedure, not only numerical parameters can be learnt, but also notational conventions such as the fonts and colors associated with specific types of objects (rivers, cities, mountains), typical prefixes or suffixes of their names (such as *r.* for river o *mt.* for mountain), etc.

An alternative way of automatic detection of such prefixes in a large map is the use of a dictionary. For each string consisting of several words, both the complete variant and the variants without the first (or last) word are to be tested. If for a specific type of objects (e.g., rivers) in most cases the string is found after taking off a specific word (e.g., "river"), then it is to be considered as notation for this type of objects.

5. SPELLING CORRECTION IN TOPONYM RECOGNITION

Due to a very complicated layout of objects and textual elements in cartographic maps, words can be recognized with errors, e.g., "RNSoSIA" for "RUSSIA" where U is erroneously recognized as N due to a nearby river, and the circle representing a city is erroneously taken for the letter o, see Figure 1. We suggest detecting and correcting such errors using the following algorithm.

1. Each string obtained from the basic OCR procedure is looked for in a list (dictionary) of expected toponyms, which (if the word is found) provides the semantic information associated with it, such as the type of object (e.g., city, river), its spatial relationships (e.g., administrative unit it is in), and its geographic coordinates if available. This information is verified using different sources of evidence, such as spatial distribution of the letters in the raster image, the coordinates of the letters, etc., as described in Section 4, and the probability of association of the string with the chosen geographic object is obtained.

2. In addition, similar strings (e.g., *RUSSIA*, *ASIA*, *Angola*, etc. for *RNSoSIA*) are looked up in the dictionary and for them, the same information is retrieved and the same check is performed, an additional source of evidence being the probability of the corresponding changes in the letters of the string, as described below.

3. The variant with the best score (probability) S_1 is considered.

4. If this best variant is good enough ($S_1 \geq \alpha$, where α is a user-defined threshold), then:
   4.1 If the score of the best variant significantly differs from the score of the second best one ($S_1 / S_2 > \beta$, β is a user-defined threshold) then this variant is accepted and is added to the database together with its associated information.
   4.2 Otherwise, human intervention is requested, and the variants are presented to the operator in the order of their scores.

5. Otherwise ($S_1 < \alpha$), no correction is applied to the recognized string. It is checked against the linguistic restrictions on the words of a given language, see Section 5.3.

   5.1 If no anomalies are found, it is considered a new toponym absent in our dictionary. It is added to the database as is and is associated with a nearby object using the algorithm discussed in the previous section.
   5.2 If an anomaly is found, the string is considered not recognized and human intervention is requested.

6. After all strings in the map are recognized, global check is performed, see Section 4.4. If this check fails, human intervention can be requested. Alternatively, the process of error correction can be repeated for this string, and then the global verification for the objects involved in the resulting changes.

As specified in Step 2, additional sources of evidence are taken into consideration when substituting a string for another similar string. Below we consider each of them.

Combination of different sources of information and not just finding the string or its spelling variant in the dictionary is important. For example, geographic information can be used to filter out the candidates that are very close to their spelling to the original string returned by the basic OCR procedure but are not located in the area in question. For instance, let the OCR procedure returned the string *Xalapa* in the area of Mexican State of Oaxaca. Such a string indeed exists in the list of Mexican cities, but the corresponding city is in the state of Veracruz. On the other hand, there is a city *Jalapa* precisely in the state of Oaxaca. Thus, it should be considered more probable that the string *Xalapa* was a result of a recognition error and that the correct string is a similar string *Jalapa*.

5.1 TEXTUAL INFORMATION

We suppose that there is available a list (dictionary) D of toponyms that can be found in a map. The list can contain much more toponyms than the map in hand—for example, all cities of the country, all seas of the world, etc. Such a list can be compiled as a combination of different sources such as governmental statistical databases, police databases, analysis of newspapers available in the Internet, etc.

For a given string s, e.g., *RNSoSIA*, a set of all strings similar to s in the dictionary D can be constructed [10]. A string s' is called similar to a string s if it differs from s in at most a certain number of the following disturbances:

- Substitution of a letter for another letter,
- Omission of a letter,
- Insertion of a letter.

With each such disturbance, a probability can be associated; in case of several disturbances, the corresponding probabilities are multiplied to obtain the overall probability of that s (*RNSoSIA*) has been obtained from s' (say, *RUSSIA*) by this sequence of errors. For the string itself ($s'=s$ if it is in D), the probability is 1.

The probabilities of the disturbances can depend on the specific letters involved, if this information is available. For instance, the probability of substitution of *I* for *J* is higher than *W* for *L*. Similarly the probability of omission of *I* is higher than that of *M*. In a cartographic map, the probability of insertion of *o* is high because of the notation for cities.

The iterative procedure described in Section 4.5 can be used to automatically adjust the model to the specific map. If the map is large or has some standard type and quality, the model can be trained by means of processing a part of the same map or another map of similar quality and manually verifying the results.

5.2 SPATIAL LETTER DISTRIBUTION INFORMATION

The distance between adjacent letters gives information on the probability of insertion or deletion type error. *Deletion-type error* (a letter is to be inserted to obtain a valid word) is highly probable if the distance between two neighboring letters is about twice larger than the average distance between the letters in the string (it can be the space between different words too). Similarly, *insertion-type error* (a letter is to be deleted from the string to obtain a valid word) is highly probable if the mean distance between the letter in question and its neighboring letters is about twice smaller than the average. Note that in these cases the corresponding correction of the word is not only acceptable but also required: the score of a string with this type of defects is decreased.

5.3 LINGUISTIC EVIDENCE

The checks described in this section are applied only to the strings not found in the dictionary for which the dictionary-based correction failed (no suitable similar string is found in the dictionary), see the Step 5 of the algorithm from Section 5. In this case, general properties of the given language can be used to detect (though not correct) a possible recognition error.

One of simple but efficient techniques of such verification is bigram (or trigram) control [11]. In many languages, not any pair (or triplet) of letters can appear in adjacent positions of a valid word. For example, in Spanish no consonant except *r* and *l* can be repeated; after *q* no other letter than *u* can appear, etc. The statistics of such bigrams (or trigrams) is easy to learn from a large corpus of texts. The multiplication of the bigram frequencies for each adjacent pair of letters in the word (and similarly for trigrams) gives a measure of its well-formedness, which can be compared with a user-defined threshold; if a bigram not used at all in the given language appears, the word is immediately marked as probably incorrect.

Other properties of words specific to a given language can be verified; e.g., in Japanese all syllables are open. If a recognized string for which no variants of correction by the dictionary are found does not pass any of the linguistic filters, it is presented to the human operator for possible correction. Note that since toponyms are frequently words of another language or proper names of foreign origin, linguistic verification can produce a large number of false alarms.

6 CONCLUSION AND FUTURE WORK

We have shown that the problem of recognition of inscriptions in the map, assigning them as names to specific objects (e.g., cities), and importing—using these names as keys—properties of these objects (e.g., population) from existing databases involves both traditional techniques of image recognition and methods specific for cartographic map processing. Our algorithm combines various sources of evidence, including geographic coordinates and object inclusion hierarchy, to choose the best candidate for error detection and correction. (In this work we focused on maps with texts. There are many maps with numerical labels—elevations, geographical coordinates, and so on. See [1], [7] for discussion on this type of maps.)

One obvious line of future development is refining the heuristics used in the discussed sources of evidence and adding new sources of evidence. For example, the basic recognition procedure can return the probability (the degree of certainness) of each letter in the string, or even a list of possible letters at the given position in the string along with their respective probabilities. The idea is that if the basic recognition procedure is certain that the letter in question is exactly the one it recognized (as opposed to just looking like this), the letter should not be changed in error correction, and vice versa.

Another issue possibly to be addressed in the future is the computational complexity, especially that of the method used to compute the integral in Section 4.2.

However, the most important line of future research are improvements to the automatic training of the statistical models, automatic learning of the notational information, and automatic determination of the parameters used in various heuristics of our method.

REFERENCES

[1] S. Levachkine, A. Velázquez, V. Alexandrov and M. Kharinov, Semantic analysis and recognition of raster-scanned color cartographic images, *Lecture Notes in Computer Science, Vol. 2390*, 2002, 178-189.

[2] D. Doermann, An introduction to vectorization and segmentation, *Lecture Notes in Computer Science, 1389*, 1998, 1-8.

[3] G. Nagy, Twenty years of document image analysis in PAMI, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(*1*), 2000, 38-62.

[4] A. Ganesan, Integration of surveying and cadastral GIS: From field-to-fabric & land records-to-fabric, *Proc 22nd ESRI User Conference*, 7-12 July 2002, Redlands CA http://gis.esri.com/library/userconf/proc02/abstracts/a0868.html

[5] 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*), 1988, 910-918.

[6] C.L. Tan and P.O. Ng, Text extraction using pyramid, *Pattern Recognition*, 31(*1*), 1998, 63-72.

[7] A. Velázquez, Localización, recuperación e identificación de la capa de caracteres contenida en los planos cartográficos. *Ph.D. Thesis*. CIC-IPN. Mexico City, 2002 (in Spanish).

[8] A. Velázquez and S. Levachkine, Text/graphics separation and recognition in raster-scanned color cartographic maps, *Proc. 5th IAPR International Workshop on Graphics Recognition (GREC 2003)*, 30-31 July 2003, Barcelona, Catalonia, Spain, 2003, 92-103

[9] R. Cao and C.L. Tam, Text/Graphics separation in maps, *Lecture Notes in Computer Science, 2390*, 2002, 168-177.

[10] A. Gelbukh, Syntactic disambiguation with weighted extended subcategorization frames, *Proc. Pacific Association for Computational Linguistics (PACLING 1999)*, 25-28 August 1999, Canada, 1999, 244–249.

[11] A. Gelbukh. A data structure for prefix search under access locality requirements and its application to spelling correction, *J. Computación y Sistemas*, 2003.

[12] R.C. Angell, G.E.Freund and P.Willett, Automatic spelling correction using a trigram similarity measure, *Information Processing & Management*, 19(*4*), 1983, 255-261.

[13] G. Hirst, A. Budanitsky. Correcting real-word spelling errors by restoring lexical cohesion, *Computational Linguistics* (to appear).

[14] S. Levachkine, Raster to vector conversion of color cartographic maps for analytical GIS, *Proc. 5th IAPR International Workshop on Graphics Recognition (GREC 2003)*, 30-31 July 2003, Barcelona, Catalonia, Spain, 77-91 (2003).

[15] A. Gelbukh and S. Levachkine, Resolving ambiguities in toponym recognition in raster-scanned cartographic maps. *Proc. 5th IAPR International Workshop on Graphics Recognition (GREC 2003)*, July 30-31, 2003, Barcelona, Catalonia, Spain (2003) 104-112.