
Extraction and recognition of geographical features from paper maps

Received: 13 September 2004 / Revised: 11 July 2005 / Accepted: 27 August 2005 / Published online: 23 February 2006
© Springer-Verlag 2005

Abstract With the emergence of Geographical Information Systems (GIS), map acquisition and recognition have become hotly pursued topics, both by the industry and the academia. The paper presents a novel methodology for the extraction and recognition of symbol features from topographic maps. The method proceeds by separating the map into its constituent layers and then attempting to recognize the features in different layers on the basis of symbol-specific geometrical and morphological attributes. Text strings have also been separated. The output is obtained in the form of an 'e-map' that is vectorized and hence is suitable for GIS. To demonstrate the usefulness of the proposed system a simple database along with a query processing facility is constructed integrating the information obtained from the e-map and 'some' user inputs. The methodology has been observed to perform quite satisfactorily.

Keywords Cartography · GIS · Feature · Layer separation · Map recognition · E-map

1 Introduction

Geographical maps have been in use since long for a variety of purposes. With computers invading every field in academics and industry, map processing is now no longer limited to paper-based maps. Efforts to automate the extraction of information from geographical maps using computers have led to the advent of Geographical Information System (GIS), which is a computer-assisted information management system of geographically referenced data. The two central problems in GIS are *map acquisition* and *map interpretation*. Though quite a lot of research has been going on in the map automation process, most researchers have focussed on raster to vector conversion, and the success in map recognition has been limited. As

such, the importance of a work in this direction cannot be overemphasized. By the term *e-map*, we mean an electronic version of a geographical map where entities are represented in such a fashion that the map can be reconstructed, updated and accessed much more easily using computer. Secondly, spatial and geographical information can be obtained as a result of query processing over this e-map.

The scanty early works on map recognition include a cartographic pattern recognition system using homogeneous parallel algorithms [4], an automatic map recognition system [5] and query-driven map recognition based on template matching [6]. Yamada et al. [7] focussed on extracting specific features such as points and lines based on a computationally intensive *multi-angled parallelism* method, whereas Samet and Soffer [8–10] reported a legend-driven map recognition system based on weighted bounded several nearest-neighbour classifier that is noise sensitive and requires separate map layers as inputs. Research has also been done on separating the layers of scanned maps by colours, as in [11, 12]. Efforts to separate text from graphics in an attempt to isolate and recognize lettering include the works in [13, 14], which are oriented more towards character separation from engineering drawing than maps. Methods for character string extraction specifically adapted to a cartographic context [15] use complex techniques of dynamic programming, graph theory and combinatorial exploration. In contrast, the method outlined here is much simple and gives good results for the maps dealt with.

The present work is an attempt to extract and recognize symbol features (henceforth is termed as only 'feature') in topographical maps. The term *feature* differs from that used in pattern recognition, but refers to the unit of data by which a geographical entity is represented. The rest of the paper is organized as follows. Section 2 gives a brief overview of the work. Sections 3 and 4 describe the layer separation and feature extraction and recognition phases in detail. Database construction and query processing is discussed in Sect. 5. Section 6 presents the results and discussion. Concluding remarks are given in Sect. 7.

2 Brief overview

For convenience, the entire work done is divided into four phases as follows.

- (1) *Acquisition*: Topographic Sheets prepared by Survey of India are scanned, giving raster images of the maps. From these images, some small regions are selected as the training images.
These images are selected so as to contain all possible features or legends, and also be sufficiently small as not to take too much processing time.
- (2) *Layer separation*: The map is separated into its constituent layers [e.g. vegetation (green), water bodies (blue), man-made structures (red) and text (black)] so that the number of features to be recognized in each layer be less, and the recognition process is made simpler.
- (3) *Recognition*: This is the most important and time-consuming phase that results in the recognition of certain features in each of the binary images produced at the end of the layer separation phase.
- (4) *E-map generation*: To make the results of recognition phase useful, as input to GIS, an *e-map*, a map recognisable by the computer, is created.
- (5) *Database construction and query processing*: Integrating the information obtained from the e-map and 'some' user inputs, a database is created, which can then be used in a manner similar to GIS to answer user queries.

A detailed description of phases 2 and 3 is given in Sects. 3 and 4, respectively. Database construction and query processing is discussed in Sect. 5.

3 Layer separation phase

The most intuitive and natural approach to this problem is to use the R, G, B pixel values for classification. However, a direct clustering approach (e.g. *K*-means algorithm) is not suitable due to wide variations in colour intensity. For example, the green component of red pixel may be more than the

green component of green pixel in the same map. Secondly, as the a priori probabilities of pixels in different layers are not comparable, the cluster corresponding to the dominant layer tends to encroach the other layers. Another problem is that the intensity values of black or white pixels have rarely the ideal values of (0, 0, 0) or (255, 255, 255) and such pixels tend to be wrongly classified. Because of the above difficulties, the strategy devised for layer separation first determines the initial cluster centres on the basis of an enhanced image, followed by the actual separation. The steps involved are described below.

Step (a). Image enhancement The training image is enhanced such that the R, G, B values of the pixels are either 0 or 255. The heuristic followed is as follows. Pixels with a small difference in their R, G, B values are set to either fully black (0, 0, 0) or fully white (255, 255, 255) depending upon the average intensity value. Otherwise the maximum of the three values is set to 255, and others to 0. So we have the following classes to represent layers: $C_0 = \{0, 0, 0\}$, $C_1 = \{255, 0, 0\}$, $C_2 = \{0, 255, 0\}$, $C_3 = \{0, 0, 255\}$ and $C_4 = \{255, 255, 255\}$ for black, red, green, blue and white, respectively. If two values are equal and the maximum, both of them are set to 255. So we have three additional classes to represent layers: $C_5 = \{255, 255, 0\}$, $C_6 = \{0, 255, 255\}$, $C_7 = \{255, 0, 255\}$ for yellow, cyan and magenta, respectively. However, since in our present experiment, selected maps do not have any such colour layers, any pixel satisfying one of these is mapped to one of C_i ($i = 0, 1, \dots, 4$) depending on the frequency of occurrences over its neighbourhood. This allows us to deal with only five classes leading to four layers.

An example of training image used in the work is shown in Fig. 1a and the image after enhancement is shown in Fig. 1b.

Step (b). Clustering The *K*-means algorithm [1] is used to generate $K = 5$ clusters, where each pixel of the original image is allocated to one of the five clusters such that the intra-class distance is minimum. The feature vector is selected to be the normalized original R, G, B values, along

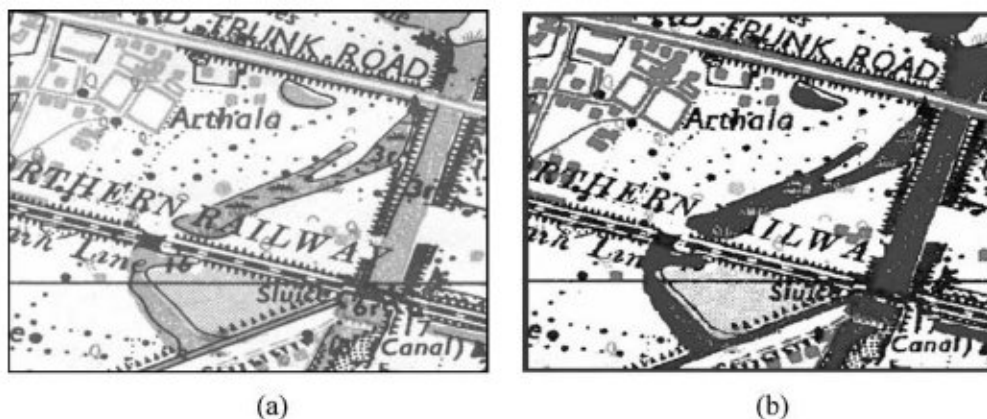


Fig. 1 a Original training image. b Training image after enhancement

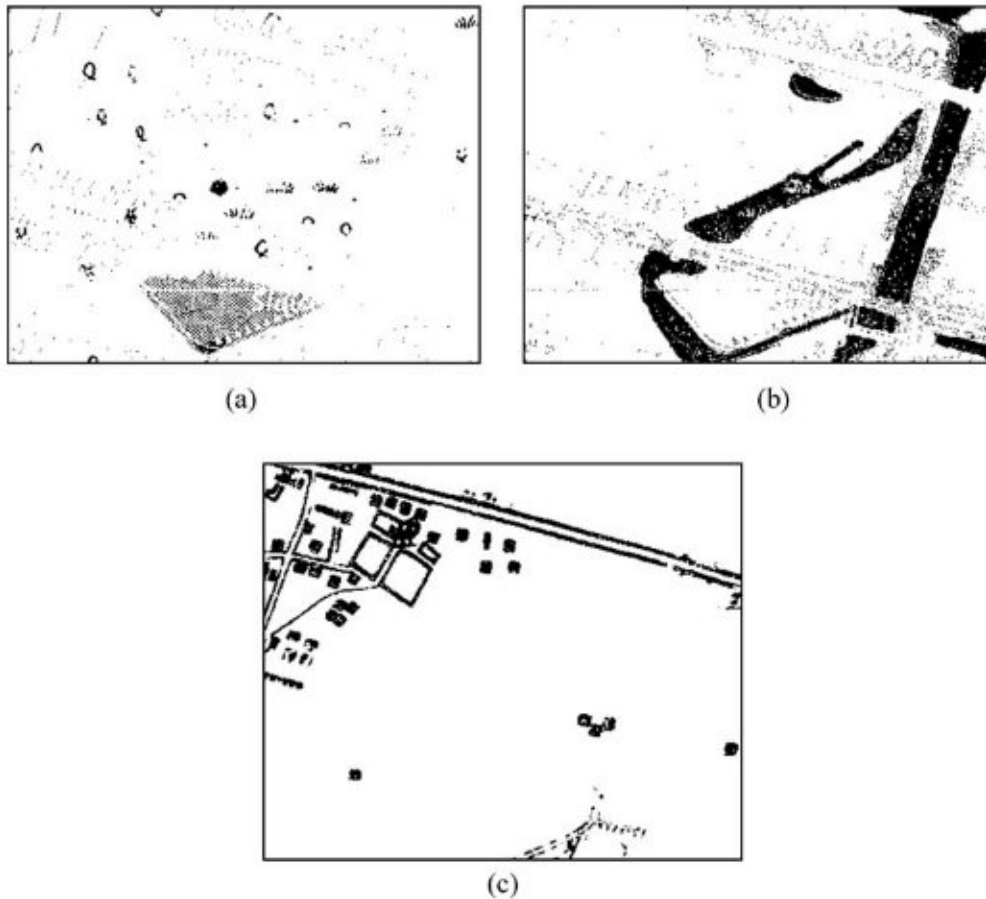


Fig. 2 a Green layer of training image. b Blue layer of training image. c Red layer of training image

with the normalized intensity I (I was added as a component of the feature vector in order to differentiate between the black and white clusters, as their normalized $\{r_i, g_i, b_i\}$ values tended to be similar) as follows:

$$r_i = \frac{R_i}{R_i + G_i + B_i}, \quad g_i = \frac{G_i}{R_i + G_i + B_i},$$

$$b_i = \frac{B_i}{R_i + G_i + B_i} \quad \text{and} \quad I_i = \frac{R_i + G_i + B_i}{3 \times 255},$$

The initial cluster centres are supplied as the average of the normalized components (r_j, g_j, b_j, I_j) of the pixels corresponding to each of the C_j 's of the enhanced image, where

$$r_j = \frac{1}{n_j} \sum_{i \in C_j} r_i, \quad g_j = \frac{1}{n_j} \sum_{i \in C_j} g_i,$$

$$b_j = \frac{1}{n_j} \sum_{i \in C_j} b_i \quad \text{and} \quad I_j = \frac{1}{n_j} \sum_{i \in C_j} I_i,$$

where n_j is the total number of points in class or group C_j of the enhanced image.

Step (c). Final separation Once the final cluster centres are determined, the map image is separated into the desired lay-

ers by a procedure similar to the *nearest prototype classification algorithm*. At the end of the algorithm, $K = 5$ binary images are generated, such that the pixel value of the i th pixel in the j th image is 1 if the feature vector of the corresponding pixel $\{f_i\}$ has been assigned to cluster C_j .

The images of separated red, green and blue layers of the training image are shown in Fig. 2.

4 Feature extraction and recognition

Even after separation of the map into its constituent layers, feature extraction and recognition remains sufficiently difficult to attract a simple solution. The procedure of template matching does not work as different instances of the same symbol vary in scale and orientation. The high level of obstruction of geographical symbols due to the map-making process causes the pixels in the intersections of two features to be put into one layer, thus making the symbols in the other layers discontinuous. For example, a bridge may cause a discontinuity in the river. Consequently, statistical pattern recognition with features invariant to scale and rotation as used in [8–10] does not work. To alleviate this problem, researchers have used images of separate map layers as inputs

to their systems. However, such separate map layers are not available to us. The quality of the print and noise in the images also add to the difficulty. Hence, the current approach focuses on 'discovering' simple spatial and geometrical attributes of the features by extensive experimentation and observation.

4.1 Feature extraction in the green layer

The features extracted and identified in the green layer include trees, grasses and green fields or forests. The steps involved are as follows.

Step (a). Noise removal In a binary image, if noise converts a black pixel to white pixel, it is considered as negative noise. Such noise usually creates holes in a region and discontinuities in strokes. Negative noise is removed by morphological closing [1]. This also causes arms of grasses and shrubs to be fused, and vacant areas within green fields to be filled, such that they would be counted as single components. Positive noise (which converts a white pixel to black

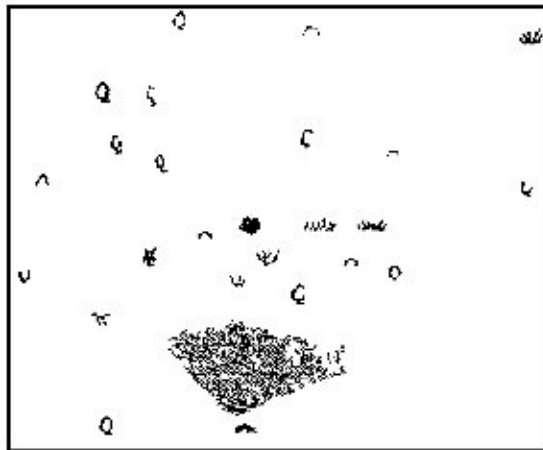


Fig. 3 Green layer after noise removal

pixel) is removed by component counting [3] and then deleting components of small size. At the end of this step, the image becomes clean (Fig. 3) and suitable for applying recognition criteria.

Step (b). Thinning or skeletonization A closer examination of Fig. 3 reveals that most of the trees have just one Y-shaped junction, whereas grasses and green fields have more of them. Thus, the number of such junctions could serve as a criterion for recognition of trees. Locating points, which have Rutovitz connectivity number equal to 3, can identify such junctions. However, to correctly identify junction points by Connectivity number, the components must be thinned. The result is shown in Fig. 4.

Step (c). Feature extraction The genus or Euler number of each remaining component $G(C)$ is computed [1]. Ideally, the Euler number of each tree should be zero as it has exactly one hole. However, because of the obstruction of geographical symbols, many of the trees do not satisfy this criterion. So, some additional criterion is required to recognize trees. As said in the previous step Rutovitz connectivity number helps identifying trees. This is computed as the number of transitions from the object pixel to the background pixel if we travel along the 8-neighbourhood of a candidate pixel of a thinned component. Then for each component, the number of pixels $n(C)$ for which this number is 3 is counted. Maximum (MAX) of the number of object-to-background transitions along any row of a component is also determined.

Step (d). Recognition Finally, identity of the components of the green layer is determined based on the following rules:

- If $G(C) = 1$ and $MAX = 2$, it is a tree.
- If $MAX = 2$ and $n(C) \leq 2$, it is a tree.
- If $3 \leq MAX \leq 6$, it is a grass.
- If $MAX > 6$, it is a green field or forest.

Figure 5 exemplifies the decision criteria. The recognized map for the green layer of the training image is shown in Fig. 6. 'T' represents tree, 'G' grass and 'F' green field.

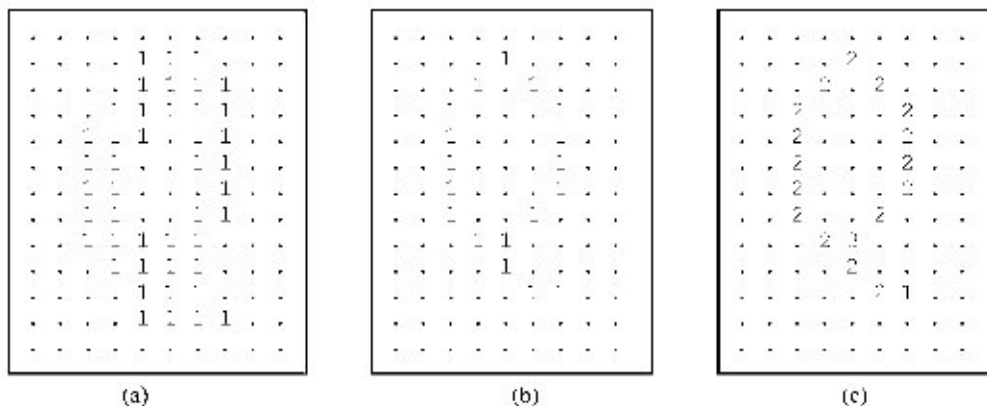


Fig. 4 a A component in the green layer of the map. b Thinned version of (a). c The same component showing the connectivity number of the pixels in the object

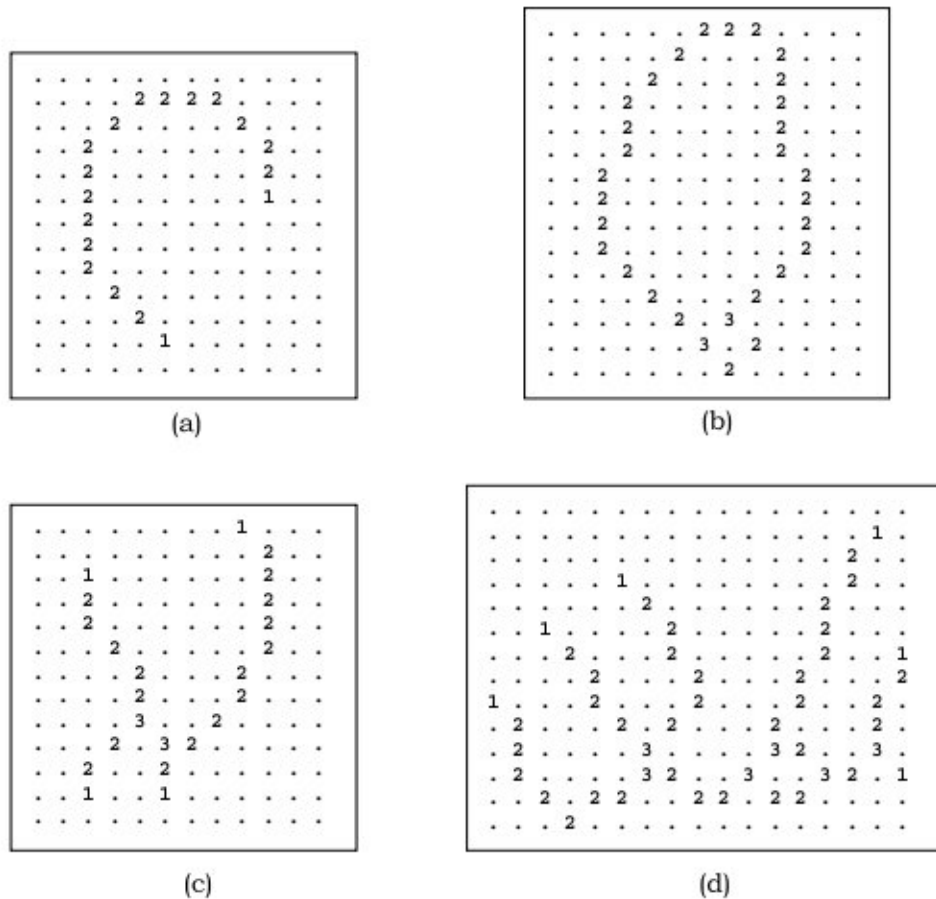


Fig. 5 **a** Example of a tree that has no hole and no point with connectivity number 3. **b** Example of a tree that has one hole and two points with connectivity number 3. **c** Example of a tree that has no hole but one point with connectivity number 3. **d** Grass

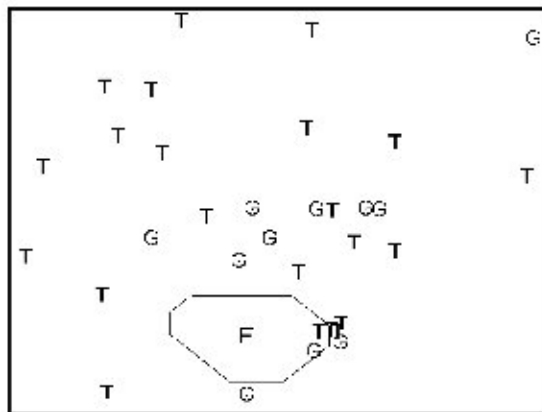


Fig. 6 Recognized green layer

4.2 Feature extraction in the blue layer

The features extracted and identified in the blue layer include rivers (R) and tanks (K). Identifying rivers is difficult, since they are intersected by bridges and dams, and thus have large parts missing from the blue layer. The different components of a river are spatially so far apart that they cannot

be joined algorithmically using any neighbourhood operator of reasonable size. Thus, human intervention seems the only way out. The steps involved are as follows.

Step (a). Noise removal The user is expected to join the components in the input image, if needed. This is followed by a component labelling step that removes positive noise. Negative noise may be removed by *median filtering* [2].

Step (b). Recognition It is done based on the following decision criteria: suppose $F_i (i = 0, 1, 2, 3)$ denote the set of pixels of one of the extreme rows or columns of the image. Then $B \cap F_i \neq \phi$ indicates that an object B intersects a side of the image. Thus,

- If $B \cap F_i \neq \phi$ & $B \cap F_{(i+2) \bmod 4} \neq \phi$ for any i , then B is a river.
- If $B \cap F_i \neq \phi$ & $B \cap F_{(i+2) \bmod 4} \neq \phi$ & $B \cap F_i \cap F_{(i+1) \bmod 4} \neq \phi$ for any i , then B is a river.
- If $B \cap F_i \neq \phi$ for all i , then B is a tank.

The e-map for the blue layer, with boundary extraction performed, is shown in Fig. 7.

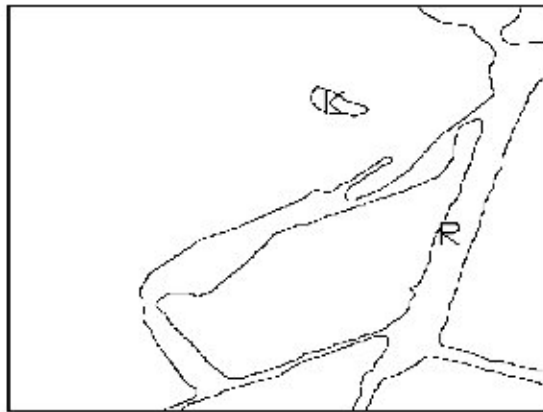


Fig. 7 Recognized blue layer

4.3 Feature extraction in the red layer

The features to be recognized in the red layer are huts (H, small rectangular blobs), metal roads (M, solid double parallel lines), cart roads (C, solid single line), non-metal roads (U, dashed double parallel lines) and village/town/human habitation (V). These feature types are processed one at a time in the order mentioned above. The detailed steps are as follows.

Step (a). Preprocessing The entire image is thinned, and the end points of the lines are identified and joined with their nearest end-point neighbour, provided that they are within a specified orientation and distance from each other [16, 17]. This joins many broken lines in which the breaks are a few pixels long. For lines with long breaks, an approach based on directional mathematical morphology [18] is used. The directions of all possible lines are first determined using Hough transform [1]. After the line joining, noise-cleaning operation is done through connected component analysis.

Step (b). Isolation of hut Huts are separated using the shape features such as (i) *aspect ratio*, the ratio of the smaller to the larger dimension of the minimum bounding box, and (ii) *vacant ratio*, the ratio of the area of the component to that of the bounding box. The components with aspect ratio and vacant ratio greater than a particular threshold (0.5 in the present case) are recognized as huts.

Step (c). Isolation of metal and cart roads The unique geometric characteristic of roads is their high length-to-width ratio or low aspect ratio. However, for roads not aligned with the x or y directions, this statement may not be true. For such cases, the vacant ratio will be quite small. Thus the components with a low aspect ratio or a low vacant ratio (less than a threshold, chosen to be 0.15 here) are recognized as metal or cart roads.

Step (d). Distinguishing metal roads from cart roads Once identified, metal and cart roads are drawn on a separate image and Hough Transform is applied to them. If in the Hough

transform domain we find two maxima at the same angle and few cells away from each other, then it represents two lines having the same inclination but only a few pixels apart. Then these are parallel lines representing a metal road. Otherwise, the roads are cart roads. Correspondingly, the lines are labelled as 'M' or 'C'.

Step (e). Detecting non-metal roads From the residual image, the components with area less than a specified threshold (chosen to be 50 here) are extracted to search for the possible presence of non-metal roads (dashed lines). Hough transform is again used on these components to determine the presence of a dashed parallel line, which represents non-metal roads. Hough transform may detect an isolated line from sporadic points, and such lines should be neglected.

Step (f). Recognizing human habitations The image left after deleting the above features consists mainly of villages or towns, barring a few small-sized components. (These components may remain because of noise.) After these small-sized components are filtered out using connected component analysis, the minimum-bounding octagon is determined for the remaining components that represent the human habitation.

Figure 8 shows the red layer after the steps mentioned above, whereas the final output of the recognition and e-map generation phase on the red layer of the training image is shown in Fig. 9.

4.4 Feature extraction in the black layer

In the black layer, text strings are to be extracted so that the names can be recognized by an OCR system and are entered into GIS. For this, the size criterion similar to that in [12] is used here to judge the components as characters. The specific steps are detailed below.

Step (a). Preprocessing Positive noise is removed by component labelling and removing small-sized components, whereas negative noise is removed by morphological closing.

Step (b). Text extraction using size criteria The size (dimensions of both sides of the bounding box) of the components is determined. A typical size histogram is expected to contain three peaks [18] when the map contains letters of single font size only (Fig. 10a). The first peak is caused by numerous specks and isolated lines, while the other two represent the width and height, respectively.

However, since the present map contains fonts of different sizes, the histogram is not exactly tri-modal, but of the shape as shown in Fig. 10b. Characters are extracted by selecting components that lie within the range $[d, e]$. The peaks within the range correspond to the height or width of different font sizes.

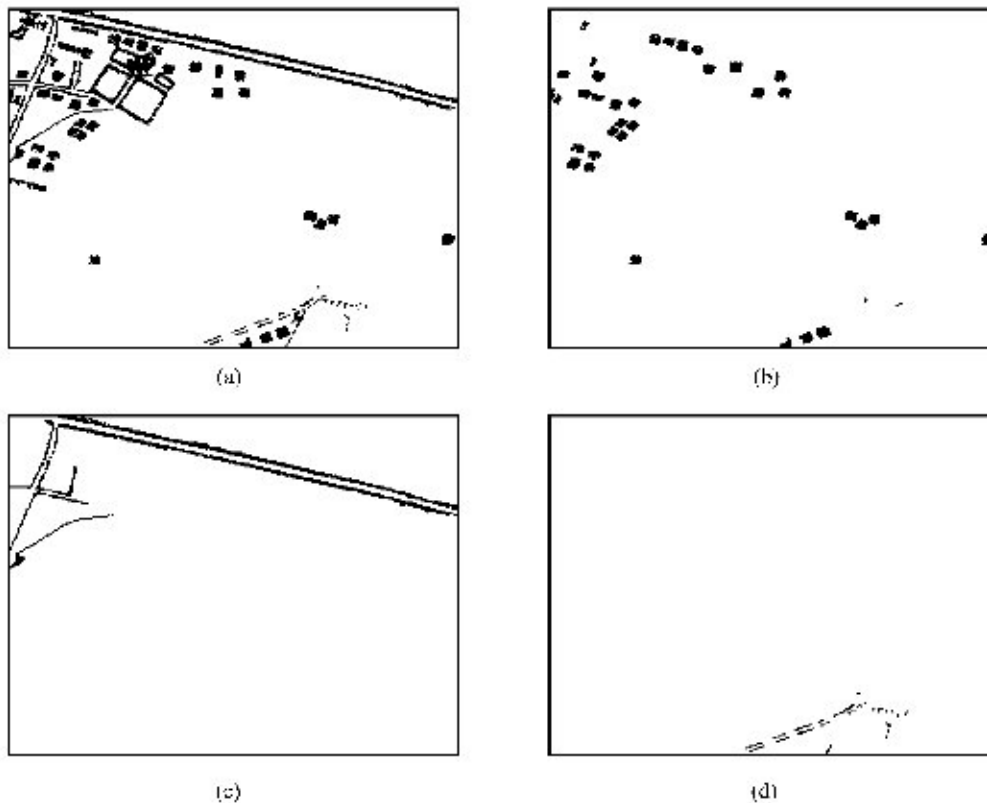


Fig. 8 Red layer of the training image. **a** After line joining and noise-removal. **b** Huts extracted from (a). **c** Metalled and cart roads isolated. **d** Unmetalled roads isolated

Other than text, some important features in the black layer include wells and tube wells. Wells are represented as circles and tube wells as triangles. The steps involved in their recognition are as follows.

Step (c). Recognition of wells The problem with recognition of wells is the presence of several different features of similar shape. Such features include telegraph lines and district boundaries. However, these features generally occur in a serial line and not in isolation. This fact is used

in the recognition of wells. The operations performed are (i) applying morphological hit-and-miss transform with circular structuring element of appropriate size and its eroded complement (this detects circles of the desired size), (ii) followed by thinning of the image (all components of circular shape would be one or two pixels in area), (iii) Hough transform of one- or two-pixel thin components to determine which of them fall in straight lines (these are not wells) and (iv) identifying one- or two-pixel area components that do not contribute to line formation in Hough transform as wells.

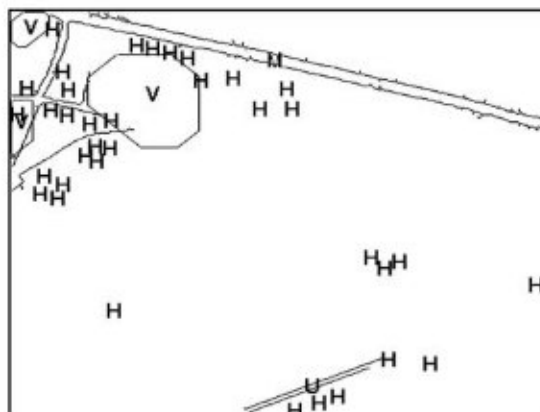


Fig. 9 Recognized red layer

Step (d). Recognition of tube wells Triangular features are easily distinguished in image by the monotonically increasing width of the projection in the direction orthogonal to the principal axis. The recognized e-map of the black layer is shown in Fig. 11.

5 Database construction and query processing

In order to make the information obtained from map recognition more useful, query-processing capability, similar to that in GIS, is added to the system. The following steps are involved.

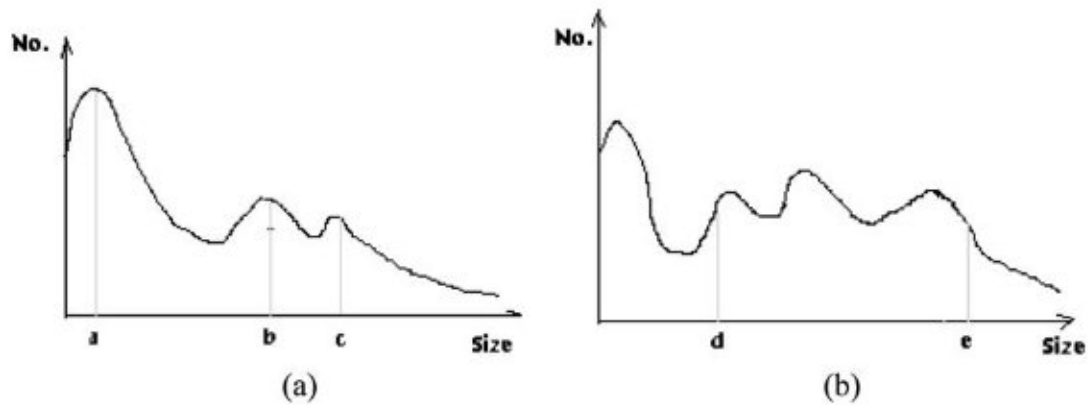


Fig. 10 a Ideal size histogram. b Actual size histogram

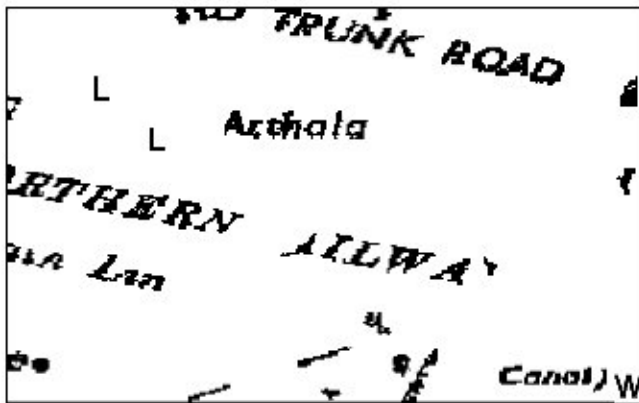


Fig. 11 Text from the black layer

5.1 Database construction

The scale of the map and the starting and ending latitudes and longitudes of the map are taken as user input. The objective is to locate any point on the map precisely giving its latitude and longitude instead of the pixel coordinates. This has the advantages that (i) it is global to all maps (pixel coordinates are local to individual maps and are dependent on image resolution), (ii) it supports query processing based on information from more than one map or parts of maps, and (iii) it enables the calculation of the actual distance between two points instead of the distance in pixels, thus being more meaningful while answering a query.

Next, the names recognized by OCR from the black layer are correlated with features detected in the other layers. This is done by matching the centre of gravities of minimal bounding box of a name in the black layer and the features in other layers. At the end of this step, the information is entered into a database in the following format: (location, feature name, proper name), where *location* is given by the latitude and longitude of the centre of gravity of a feature, *feature name* is the feature recognized by the current system and *proper name* is the name that is found to be the OCR-retrieved name associated with the feature. Note that one field of the above triplet may remain empty.

Some extra processing had to be done for 'extended' features such as rivers and roads. For roads, the locations of the end points are stored instead of a single point. This is sufficient since roads are approximated by straight lines and reconstructed by Hough transform and hence their end points are known. The storage of river information is trickier since the above statement is not true for rivers. Hence, the following steps are performed:

- (i) Determination of the medial axis of the river by thinning (this is done before the edge detection is done for the final e-map output.)
- (ii) The curve so obtained is approximated piecewise by straight lines using the split-and-merge algorithm [21]. The end points of the 'polylines' so obtained are stored. Widths of the river at these end points are also stored such that width at intermediate location can be determined.

Examples of entries in our database include $((x_1, y_1)$, forest); $((x_{21s}, y_{21s}), (x_{21e}, y_{21e}), (x_{22s}, y_{22s}), (x_{22e}, y_{22e})$, river, Ganges); $((x_3, y_3)$, railway station, Dharampur); $((x_{4s}, y_{4s}), (x_{4e}, y_{4e})$, metalled road, G T Road) and $((x_5, y_5)$, Behrampur).

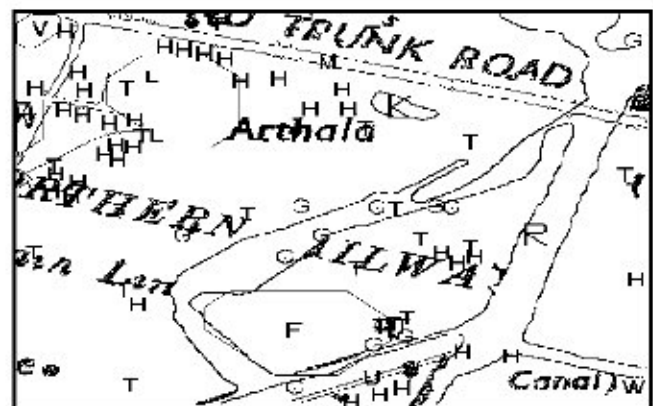


Fig. 12 Final e-map of Fig. 1a

5.2 Query processing

At present the following query functions are supported:

- *dist(source location/proper name, destination location/feature name/proper name)*

This provides the distance between two locations, for example, *dist(Behrampur, river)*. There might be more than one output in case one of the entries is a feature name. However, the output is undefined if both the entries are feature names, e.g., *dist(forest, metalled road)*.

- *list(source location/proper name, destination feature name/proper name, distance)*

This provides an ordered list of locations (along with distances) that may be found within a specified distance of the source location, for example, *list(Gazipur Dairy Farm, tank, 10 km)*. This function may also be used to

determine if a certain feature is at all present within a specified distance of a location by checking whether the output is null.

As in database construction, extra work needs to be done in the case of extended features for query processing too. For roads, since the end points are known, finding the distance from a point location involves calculating the perpendicular distance from the point to the straight line. For rivers that are approximated by multiple straight lines, determining the distance from a compact feature involves determining which line segment is closest (i.e. which consecutive end points are closest) to the point location, and then calculating the perpendicular distance from that line segment to the point. For the *list* query, the location output by the system corresponds to this 'closest point' on the extended feature to the compact feature.

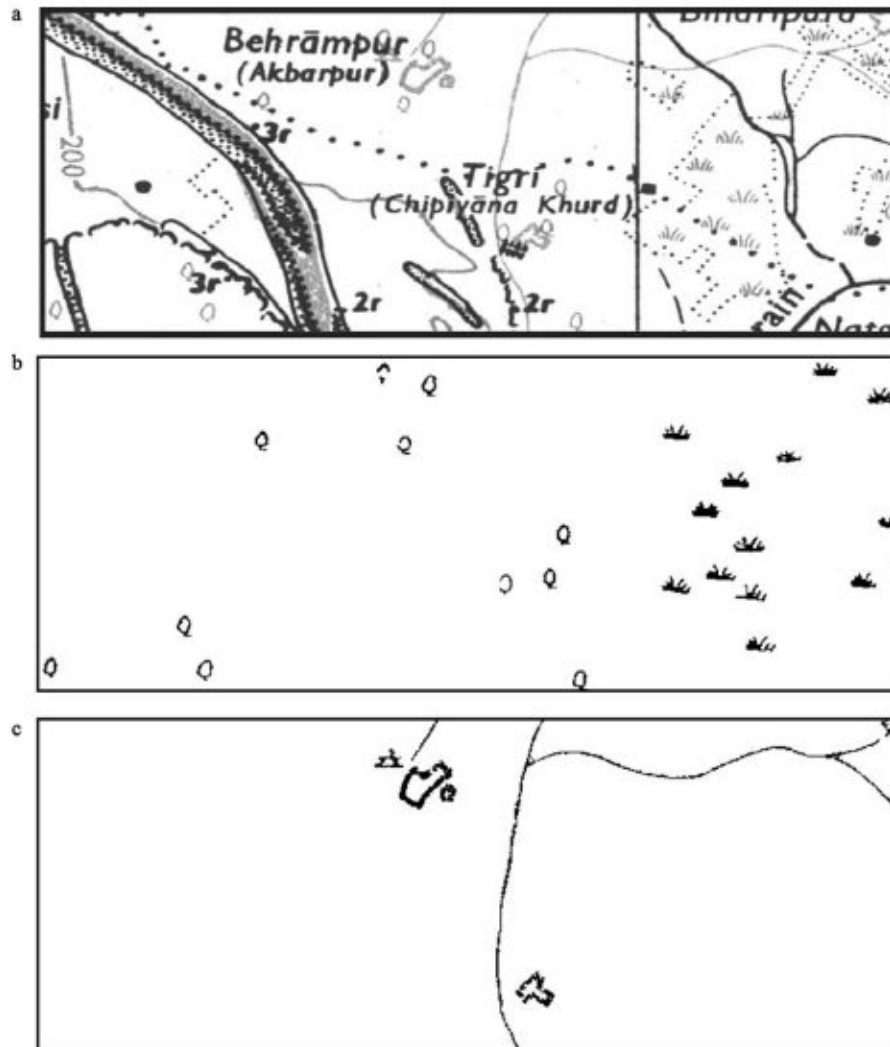


Fig. 13 Results of the proposed methodology on test image I. **a** Original image. **b** Green layer after noise cleaning and closing. **c** Red layer after noise cleaning and line joining. **d** Blue layer after preprocessing. **e** Black layer after noise—cleaning. **f** Final e-map

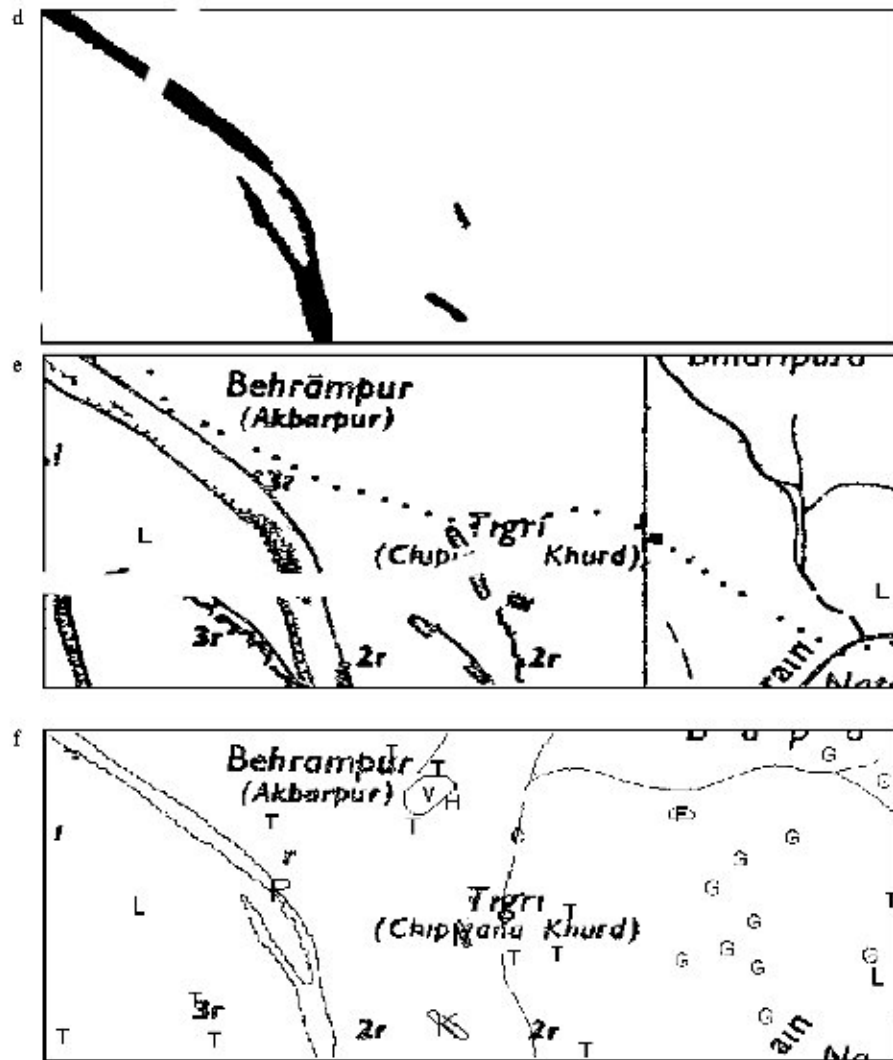


Fig. 13 Continued

6 Results and discussion

6.1 Results from the training image

The training image was taken from the Survey Map 53H/6 showing parts of Delhi, Ghaziabad (UP) and Faridabad (Haryana) prepared by the Survey of India. The map was scanned at 300 dpi, giving a $2,024 \times 2,035$ pixel image. From this map, the training image of size 400×380 pixels was chosen. The final e-map for this image (after combining the outputs of the different layers) is shown in Fig. 12. The facility used for this purpose was a Silicon Graphics machine with IRIX 5.6 as the operating system.

For evaluating the performance of the proposed method the said map is manually ground-truthed. A comparison of the number of features in the actual map and that in the e-map is given in Table 1. The result of this count shows a high recognition rate of 90%. However, it should be noted that

the system is tested only on a handful of images obtained by scanning SOI maps of the Indian Northern Plain region. So no general conclusion can be drawn immediately. In spite of this fact, it should be mentioned that results obtained by such low-cost features and simple rules are encouraging and suggest further exploration in this direction. Most of the failures are resulted from the fact that recognition of all symbols was not attempted, and hence such features were interpreted wrongly.

6.2 Count of errors

Three types of errors [13] are common in pattern recognition, optical character recognition (OCR) and document image analysis, and serve as an objective measure for the evaluation of the adopted methodology adopted. These are as follows.

- (i) *Substitution errors*—a valid label is assigned incorrectly to a valid input symbol.

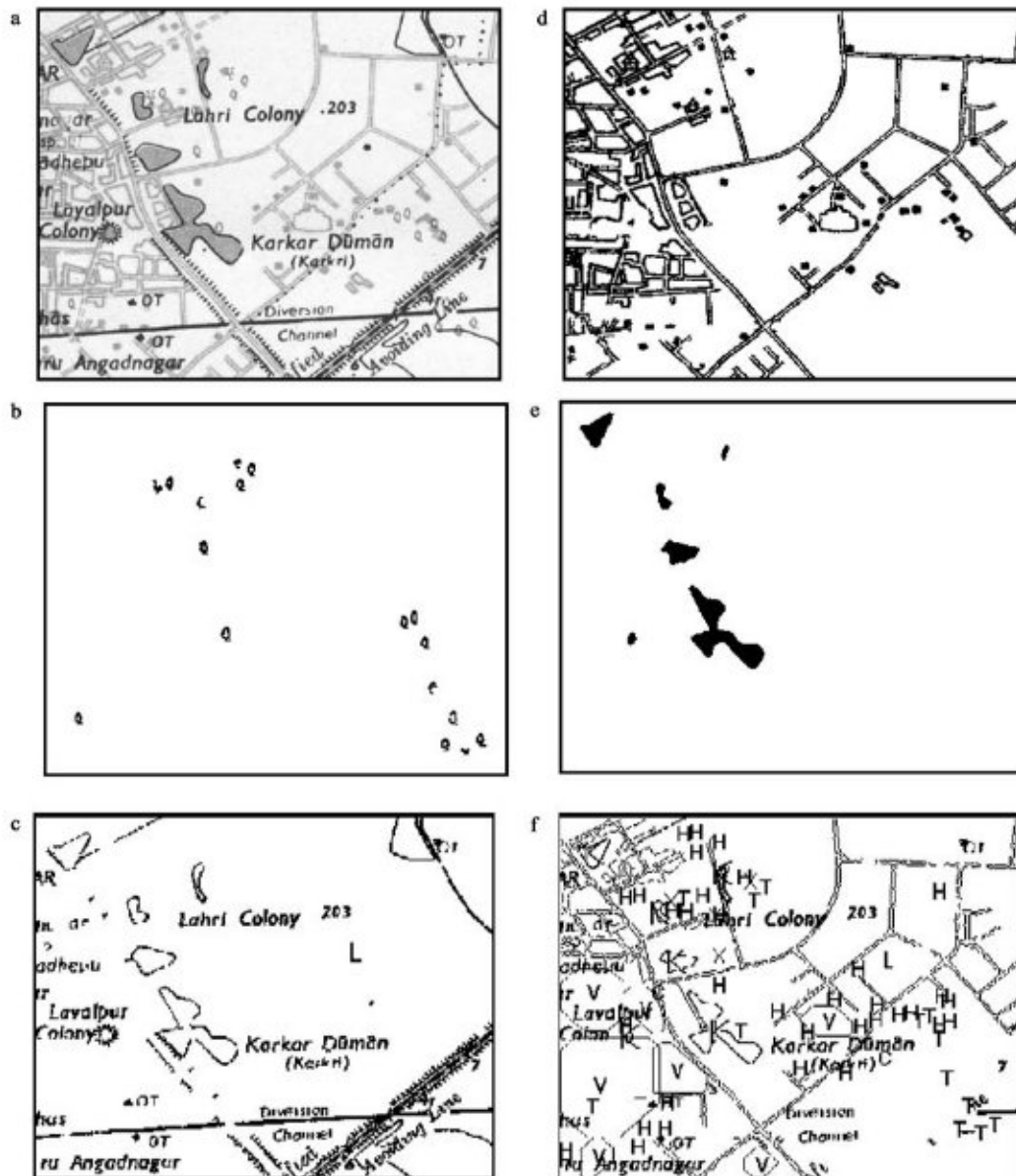


Fig. 14 Results of the proposed methodology on test image II. a Original image. b Red layer after preprocessing. c Green layer after closing. d Blue layer after noise—cleaning. e Black layer. f Final e-map

- (ii) *Deletion errors*—a valid symbol is classified as unknown or undefined.
- (iii) *Insertion errors*—an invalid symbol is classified as one of the valid symbols.

Using the above evaluation criteria, the following results are obtained on the training image: 1 substitution errors (one tree was wrongly identified as grass), 1 deletion errors (one tree could not be recognized) and 2 insertion errors (two components which are actually parts of a dam were recognized as huts). This count does not include the errors of counting four trees and three grasses for components that are actually part of the green field but recognized sepa-

rately because they remained as separate components in the image.

6.3 Results for other test images

The map recognition procedure is performed on many other test images, all drawn from the same map as is the training image. The results of three of them are shown in Figs. 13–15. The count of errors for these images too is small, and is displayed in Table 2. In all the three cases, recognition rates in excess of 90% have been recorded. Secondly, the results do not differ up to addition of 10% Gaussian noise.

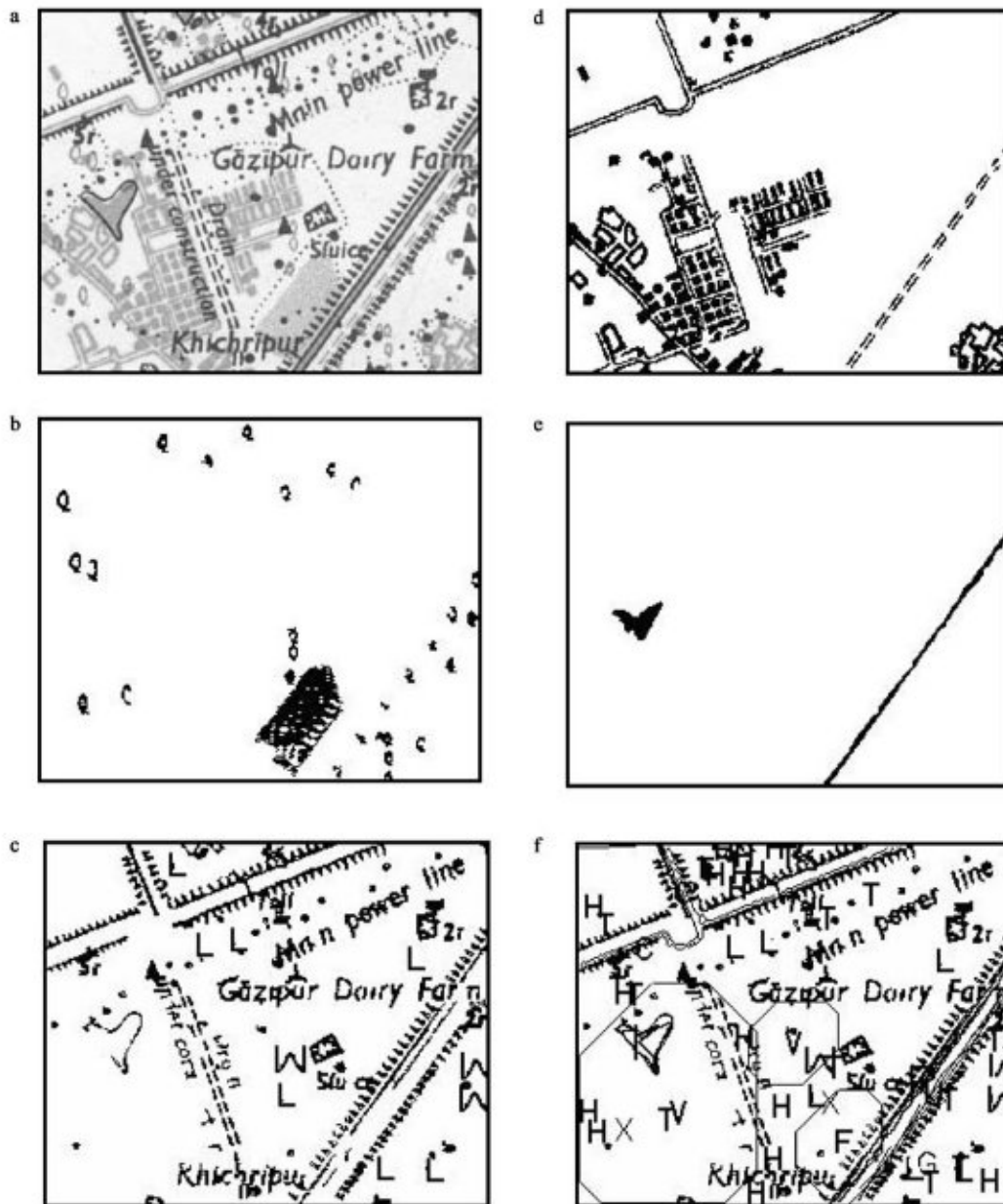


Fig. 15 Results of the proposed methodology on test image II. a Original map. b Red layer. c Green layer. d Blue layer. e Black layer. f Output e-map

The low count of errors and the robustness to noise show the methodology followed in particularly good light.

6.4 Query processing results

The results of query processing are sufficiently accurate for queries involving 'compact' features such as trees and tanks, and mixed for features such as rivers, roads and railway lines. This is understandable since the 'centre of gravity' of these features often does not make a true representation of location of the feature.

Some sample queries and their answers on our system are as follows:

Query: `dist(Arthala, Grand Trunk Road)`
 Result: Source = Arthala (28 deg 40.4 min lat, 77 deg 23.7 min long); Destination = Grand Trunk Road (28 deg 40.56 min lat, 77 deg 23.7 min long); Distance = 0.25 km.

Note. See training image.

Query: `list(Behrampur, well, 2)`
 Result: Reference: Behrampur (28 deg 38 min lat, 77 deg 24.2 min long), Feature: well. List: (i) At (28 deg 37.7 min lat, 77 deg 23.9 min long) distance 0.75 km, (ii) At (28 deg 37.9 min lat, 77 deg 25.5 min long) distance 2 km.

Note. See Test Image I.

Table 1 Comparison of features in the actual map and in the e-map of training image

Feature	Number present in actual map	Number present in e-map
1. Tree	18	20
2. Grass	8	11
3. Green field	1	1
4. River	1	1
5. Tank	1	1
6. Hut	34	36
7. Metal road	1	1
8. Non-metal road	1	1
9. Cart road	0	0
10. Village/town	3	3
11. Well/tubewell	3	3

Table 2 Count of errors for the test maps

Error type	Test image I	Test image II	Test image III
Substitution errors	1	1	1
Deletion errors	0	4	4
Insertion errors	0	0	0

The entries in our database have been inserted as a chained hash table. Each bucket of the hash table is indexed by a separate feature. Chains for each feature contain the actual entries sorted on the location (latitude first). While this is a sufficiently simple data structure providing quick insertion and also quick search for queries containing a feature name, it is not always the best data structure for all types of queries in terms of search time. However, it is adequate for the purpose of demonstration of our query-processing ability and a better performing data structure and/or algorithm may be added at any point of time.

7 Conclusion

In this paper we have presented initial steps of development of a prototype system for automatic recognition of map symbols from the standard paper maps for generating e-map. The system puts together mostly well-known methods and makes them work in particular order to achieve the said goal almost without any user intervention. So the novelty of the work lies in selection and engineering of the appropriate tools. The proposed heuristics are seen to be robust and perform well in the presence of noise. Obviously, the success of the system relies on the success of each tool at various steps. For example, success of layer separation depends on the ability of the clustering algorithm. There are a good number of clustering algorithms available in the literature, and we have used one of the simplest ones, namely *K*-means. With its known limitations (e.g. it detects isotropic clusters around its centre), it can do its job satisfactorily given pre-computed seed points. This is because colours used in maps are supposed to be uniform (i.e. without any shading) except noise due to paper and print quality and also due to digitization. Secondly, in road extraction stage, selection of

structuring element for directional morphological operation, ample care should be given. Though some default values are incorporated in the system for this as well as some other parameters, sometimes user intervention is needed to produce more fruitful results. An AI-based technique for the selection of such values seems to be helpful and we are searching for a suitable one.

The results of the preliminary query-processing system have been very encouraging, and efforts are on to serve more complicated and useful queries. For example, the distance between a linear feature, e.g., a road or a railway track, and a compact feature may be more useful than the Euclidean distance between two locations in many cases. Again, more work needs to be done in tackling problems while building a composite database from more than one map, since the same object, for example a river or a road, might be present in more than a map.

References

- Chanda, B., Dutta Majumdar, D.: Digital Image Processing and Analysis. Prentice Hall, New Delhi (2000)
- Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Addison-Wesley, Reading, MA (1999)
- Rosenfeld, A., Kak, A.C.: Digital Picture Processing, 2nd edn., vol. 2. Academic Press, New York (1982)
- Starr, L.E.: Computer-assisted cartography research and development report. International Cartographic Association (1984)
- Ejiri, M., Kakumoto, S., Miyatake, T., Shimada, S., Ichikawa, T.: Automatic recognition of design drawings and maps. In: Proceedings of the Seventh International Conference on Pattern Recognition, pp. 1296–1305 (1984)
- Kasturi, R., Alemany, J.: Information extraction from images of paper-based maps. IEEE Trans. Softw. Eng. **14**(5), 671–675 (1988)
- Yamada, H., Yamamoto, K., Hosokawa, K.: Directional mathematical morphology and reformalized hough transformation for analysis of topographic maps. IEEE Trans. Pattern Anal. Mach. Intell. **15**(4), 380–387 (1993)
- Samet, H., Soffer, A.: A legend-driven geographic symbol recognition system. In: Proceedings of the Twelfth International Conference on Pattern Recognition, vol. 2, pp. 350–355, (1994)
- Samet, H., Soffer, A.: MAGELLAN: map acquisition of geographical labels by legend analysis. Int. J. Doc. Anal. Recognit. **1**(2), 89–101 (1998)
- Samet, H., Soffer, A.: MARCO: map retrieval by content. IEEE Trans. Pattern Anal. Mach. Intell. **18**(8), 783–798 (1996)
- Lauterbach, B., Anheier, W.: Segmentation of scanned maps into uniform colour spaces. In: IAPR Workshop on Machine Vision Applications, pp. 322–325 (1994)
- Ebi, N., Lauterbach, B., Anheier, W.: An image analysis system for automatic data acquisition from coloured scanned maps. Mach. Vis. Appl. **7**(3), 148–164 (1994)
- Fletcher, L.A., Kasturi, R.: A robust algorithm for text string separation from mixed text/graphics images. IEEE Trans. Pattern Anal. Mach. Intell. **10**(6), 910–918 (1988)
- Gao, J., Tang, L., Liu, W., Tang, Z.: Segmentation and recognition of dimension texts in engineering drawings. In: Proc. 3rd Int. Conf. on Document Analysis and Recognition, vol. 1, pp. 528–531 (1995)
- Deseilligny, M.P., Men, H.L., Stamon, G.: Characters string recognition on maps: A method for high level reconstruction. In: Proc. 3rd Int. Conf. on Document Analysis and Recognition, vol. 1, pp. 249–252 (1995)

-
16. OpenGIS Consortium: The OpenGIS Abstract Specification (1996), www.opengis.org/techno/abstract.html
 17. Man, V.D., Gerland, P.: Software development and recent technological innovations for geographical and statistical applications for population activities—the UN statistics division experience. 1st Int. Conf. on Health Geographics (1998), <http://sauvy.ined.fr/seminaires/demodynamiques/98-99/gerland-text/jhupaper.html>
 18. Koundinya, K.N.V.L.N., Chanda, B.: Detecting lines in graylevel images using search techniques. *Signal Process.* **37**, 287–299 (1994)
 19. Koundinya, K.N.V.L.N., Chanda, B.: Line detection using search techniques and list processing. *Int. J. Syst. Sci.* **26**, 1053–1077 (1995)
 20. Agam, G., Dinstein, I.: Adaptive directional morphology with applications to document analysis. In: Maragos, P., Schafer, R.W., Butt, M.A. (eds.), *Mathematical Morphology and its Applications to Image and Signal Processing*, pp. 401–410. Kluwer, Boston (1996)
 21. Mukherjee, D.P.: *Fundamentals of Computer Graphics and Multimedia*. Prentice Hall, New Delhi (1999)