To The Chief Librarian

LIBRARY COPY

Deeptendu Bikash Dhar
(DEEPTENDU BIKASH DHAR)
M. Tech. (CS)
2002

M.Tech. (Computer Science) Dissertation Series

FEATURE EXTRACTION FROM MAPS

A dissertation submitted in partial fulfillment of the requirements for the Master of Technology (Computer Science) degree of the Indian Statistical Institute

By

DEEPTENDU BIKASH DHAR

Under the supervision of

PROF. BHABATOSH CHANDA, ECSU

INDIAN STATISTICAL INSTITUTE

203, Barrackpore Trunk Road

Kolkata – 700 108

2002

CERTIFICATE OF APPROVAL

This is to certify that the thesis entitled "***Feature Extraction From Maps***" is an authentic record of the dissertation carried out by **Deeptendu Bikash Dhar** under my supervision and guidance. The work fulfils the requirements for the award of the M.Tech. degree in Computer Science.

Dated : The 8th of July, 2002.

B. Chanda

Signed :
(Prof. B. Chanda)
Supervisor

P.K. Nandi 8/7/2002

Countersigned :
External Examiner

ACKNOWLEDGEMENTS

It has been a great honour and a rewarding experience to work under the auspices of a guide as eminent as Prof. Bhabatosh Chanda, ECSU. Endowed with the amazing capability of making complex mathematical expressions and impossible-to-understand theories seem as simple and interesting as Enid Blyton stories, it has been Prof. Chanda's persistent efforts and encouragement that has borne fruits in the successful completion of this project. No amount of thanks can repay his contribution to this work.

A special note of thanks goes to Dr. M.K.Kundu, MIU for his constructive suggestions and criticisms, and to Dr. D.P.Mukherjee, ECSU for his prompt help in making journal papers from the IEEE Digital Library available to me through his e-mails from US.

I am fortunate enough to be associated with an Institute of international repute that has provided me excellent infrastructure and environment in helping me complete my project. I take this opportunity to thank all staff and Research Scholars of ECSU whose presence allowed me to work freely at late hours and Saturdays.

Deeptendu Bikash Dhar
July 8, 2002
Indian Statistical Institute, Kolkata

Abstract

With the emergence of Geographical Information Systems (GIS) as a means for representing and storing Cartographic data on the computer, Map Acquisition and Map Recognition have become hotly pursued topics, both by the industry and the academia. The present report gives an account of a novel methodology followed in the extraction of features and their recognition 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 feature-specific geometrical and spatial attributes. Text strings have also been separated. The output is in the form of a 'E-Map' that is vectorized and hence suitable for use by GIS. The methodology has been observed to produce recognition rates in excess of 90%.

Index Terms: *Cartography, GIS, feature, Layer Separation, Map Recognition, E-Map.*

CONTENTS

****

Chapter 1

INTRODUCTION

1.1 Background and Motivation

Man's use of maps has a long and remarkable history. From the first known map engraved on a clay tablet over four thousand years ago by the Mesopotamians, to today's comprehensive atlases, man has shown an increasing reliance on information contained in maps. Today, maps are used for a variety of information needs.

Despite the evolution of map use, the means of representing geographical information has remained unchanged to paper-based 'maps till recently. Usually, to update a map or show the results of changes made to the data, a new map must be drawn, making the whole process time-consuming and very laborious. However, with the IT revolution and the overwhelming success of the Internet, there has been attempts to computerise everything man can think of. Since computers can process large amounts of data and perform repetitive and tedious tasks with great precision, and because of the large amount of information available in maps and the frequency and variety of their use, it is natural that efforts be made to automate the extraction of information from geographical maps using computers. The consequence of such activities has been the emergence of Geographical Information System (GIS).

GIS is basically a computer-assisted information management system of geographically referenced data. It supports spatial decision-making and is capable of linking descriptions of location (spatial data) with the characteristics of the phenomena (attribute data) found there. The spatial data contain information in the form of digital co-ordinates, usually from maps or remote sensing. These can be points (for example villages or pine trees), lines (roads or railway lines) or polygons (districts or other boundaries). The attribute data contain information about the characteristics of the spatial features, for example, population of a village, type of railway line, or name of districts. GIS can address questions on locations, conditions, trends, routing and patterns, and can also simulate 'what if' scenarios. In short, GIS has become a vital tool to help nations to understand what resources they have and how best to use them.

However, GIS requires map data to be stored in the computer, along with their associated information. Unfortunately, conventional methods of storing maps as raster images will not work, not only due to the sheer volume of the data to be stored, but also due the problem of interpreting the data. Thus GIS stores data in vector format. One of the central issues in GIS is ***Map Acquisition*** – finding ways to convert data that is stored on maps to a representation compatible with GIS. By far, the most common means of this conversion is by means of a digitizing tablet. For example, we might digitize a country boundary in vector form by mounting a map on a digitizing table, capturing the locations of points along the boundary, assuming that the points are connected by straight line segments, and then produce an ASCII file of pairs of (x,y) coordinates as an approximation of the actual boundary. Recently, optical scanners have been put to use for this purpose. The maps are first scanned; the raster data is then converted to vector data with heavy user intervention in order to assure the quality of this conversion.

While supporting the map acquisition process is important, it is equally useful and even more challenging to automate the ***Map Interpretation*** process in order to locate geographic objects and their relations. Indeed information given by map legends or given as basis of data models GIS is often insufficient to recognize not only geographical objects relevant for a certain application, but also *patterns* of geographical objects which geographers, geologists and town planners are interested in. Map interpretation tasks such as the detection of morphologies characterizing the landscape, the selection of important environmental elements, both natural and artificial, and the recognition of forms of the territorial organization require abstraction processes and deep domain knowledge that only human experts have.

Although quite a lot of research has gone in the field of automating the conversion, 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 overemphasised.

The present work is an attempt to extract and recognize features in topographical maps in a manner that will be useful to GIS. The term ***feature*** differs from that used in Pattern Recognition, but relates to the unit of data by which a geographical entity is represented in computer systems and, according to the OpenGIS Consortium (OGC) terminology is modelled through a series of properties [21]. Such a work encompasses fields as diverse as Cartography, Image Processing, Pattern Recognition, Machine Vision and Artificial Intelligence and is an enormous challenge to any individual or group working in the area. Thus the current project is only a humble endeavour in the direction, with the focus more on correct recognition of a limited number of symbols, rather than a faulty recognition of all symbols found in the map legend.

1.2 A Brief Survey of Related Work

Although several works in the field of map data processing has been reported in literature, most researchers have focussed on raster-to-vector conversion [4, 5]. However, some works on Map Recognition have also been done. In [6] a cartographic pattern recognition system using homogeneous parallel algorithms and application of digital raster methods on a road-river network has been described. Ejiri *et al.* have described an automatic map recognition system [7]. Some works have also been done on feature extraction from vectorized topographic maps, as in [8], where first order rule induction on pattern recognition has been used.

In [9] 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 query requesting information from the geographical area that corresponds to this portion of the map (e.g., city name). One part of this process involves symbol recognition, such as identifying the symbol representing a city. The process used to identify this symbol is not described in detail. It is most likely tailored towards a specific map and based on template matching.

Some works have also focussed on extracting specific features from maps, such as contour lines, as in [10], or points and lines, as in [11]. The extraction in [11] is based on a method called *Multiangled Parallelism* (MAP). The MAP operation performs parallel calculations on directional feature planes. Symbols are extracted using a reformalized parallel version of the Hough Transform on these directional planes. The method is computationally intensive. Running on a dedicated image processor, it takes 18 minutes to perform the first step for a 640×512 image. Several more minutes are required for feature extraction for each symbol. While the results presented using this method seem robust and do not require separate layers, the time required to process each image make this process difficult to use for a large collection of maps.

In [12], Samet and Soffer have described a legend-driven map recognition system that is claimed to achieve recognition rates of 95%. They classify symbols based on weighted bounded several nearest neighbour classifier. This system have been extended in [13] in a system named MAGELLAN, that utilizes the symbolic knowledge found in legends to drive symbol recognition, and in [14] in a system named MARCO, that can be used for the acquisition, storage, indexing, and retrieval of map images. However, to avoid the problem due to obstruction of different map layers, they have used separate map layers as inputs to their system, which is not readily available. Thus their system fails on composite coloured maps and is also sensitive to noise.

Research has also been done on separating the layers of scanned maps by colours, as in [15] and [16]. In comparison, the method adapted here has been much simpler and robust.

Finally, mention must be made of efforts to separate text from graphics in an attempt to isolate and recognize strings or letterings in maps. The first few steps of most of these works involve finding connected components and Hough Transform (or some other collinearity criteria). In [17], the development and implementation of an algorithm for automated text string separation that is relatively independent of changes in font style, size and orientation is described. However, the works in [17,18] are oriented more towards character separation from engineering drawing than maps. Luo *et al.* [19] have used an approach based on directional mathematical morphology. However, this work emphasizes more on restoration of broken lines than the character separation; it assumes that the maps consist of only linear features and text. Thus they obtain the characters by subtracting the restored lines from the original. Deseilligny *et al.* present a method for character string extraction that is specifically adapted to a cartographic context. They emphasize a high-level reconstruction process that resolves the ambiguities remaining from pattern analysis and structure the characters into strings. The method implemented for solving this problem uses different techniques such as graph theory, dynamic programming, and combinatorial exploration.

The above efforts are primarily academic. In [22], a software called PopMap and its companion MapScan has been described. The UN Statistics Division in collaboration with the Vietnam Institute of Information Technology developed the first version of PopMap in 1989. PopMap was designed not simply as mapping software, but as an integrated and structured database of statistics and maps. It offers an integrated geographical database for statistical data with desktop mapping/GIS software facilities.

MapScan, developed in 1994, is a software package for automatic map data entry. It accepts various formats of scanned maps or drawings, reads and converts them into vector maps in formats that can be used by popular GIS/mapping systems.

In addition, there are several companies that provide scanning services that include raster-to-vector conversion. SMARTSCAN, Audre and Scangraphics are among the companies in this field. Unfortunately, the systems that these companies use are proprietary and the research involved in their system is unpublished. The products delivered by these companies are usually very accurate. These excellent results are attributed to a combination of manual editing of 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.

1.3 Scope of the current work

Before proceeding further, it is worthwhile to emphasize that the approach in the current work is to recognize different geographical features and also separate text from the map, based on their geometrical/spatial properties as determined by extensive experimentation and observation. The method works on composite topographical maps and proceeds after separating the different layers. As a result the approach adopted here is different from the works mentioned above and is also reasonably simple.

The rest of the report is organized as follows. Chapter 2 gives a brief overview of the work. Chapters 3 and 4 describe in details the two most important phases of the project, viz., Layer Separation and Feature Extraction & Recognition. Throughout these chapters, the output of each step on a 'training image' is shown. Chapter 5 presents the results, first of the training image, and then of some other maps on which the algorithm has been tested. Chapter 6 contains concluding remarks, including scope for improvements and further work in this field.

****

Chapter 2

BRIEF OVERVIEW OF THE PROJECT

2.1 Introduction

For purposes of convenience, the entire work done can be divided into four phases :

1. Acquisition Phase.
2. Layer Separation Phase.
3. Recognition Phase.
4. E-map Generation Phase.

A brief description of the phases is given below. A more detailed description of phases 2 and 3 will follow in the later chapters.

2.2 Acquisition phase

Survey Sheets prepared by Survey of India are scanned, giving raster images of the maps. From these images, a small area is selected as the training image. This image is selected so as to contain a maximum number of different features/ legends, and also be sufficiently small as not to take too much processing time.

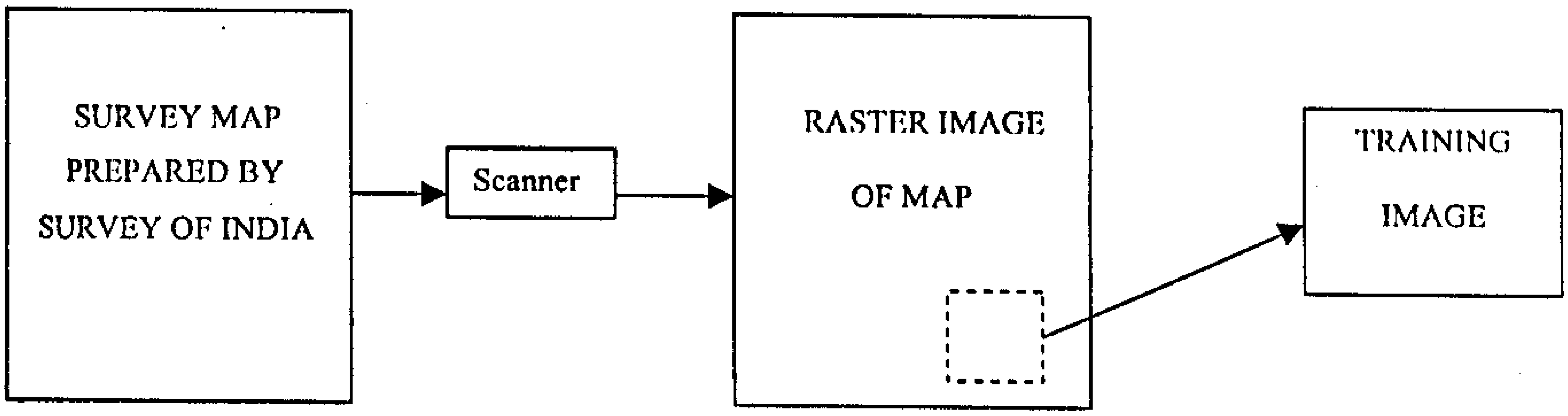


Figure 2.1 – Schematic view of the Acquisition Phase.

2.3 Layer Separation

A map is composed of several layers – layers of vegetation, water bodies, man-made features, contour lines, and the like. These layers are prepared individually, and while composing, they are given different colours for the purpose of visual clarity, as for example, green for vegetation, blue for water bodies, red for man-made structures, brown for contour lines, and black for letterings and certain other features. Thus, before attempting to recognize map symbols, if these layers could be separated out, the number of features to be recognized in each layer would be reduced, and the recognition process would become simpler.

To this end, a two-phase clustering – classification algorithm is used for the training image. The results of the clustering on the training image could be used for all other subsequent images scanned from the same map (or even different maps) to classify them directly in one stage. The procedure is detailed in the following chapter. It is to be mentioned here that the training image and other images taken from the scanned maps for testing represented essentially plane areas, and no contour lines are found there. Therefore layer separation has been done for red, green, blue and black layers. The end results of the layer separation phase are four binary images corresponding to these layers in the original image.

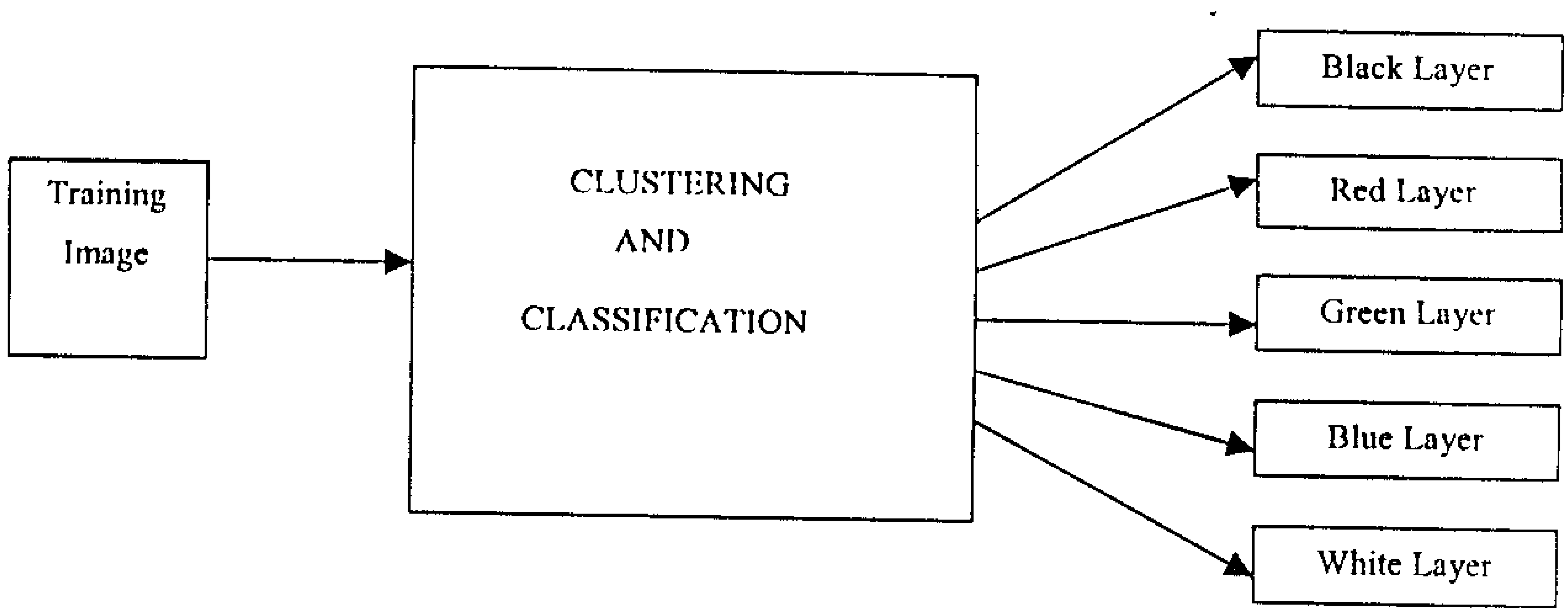


Figure 3.2 – Schematic view of the Layer Separation Phase.

2.4 Recognition phase

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. It includes applications of various well-known Image Processing algorithms such as Connected Component Labelling, Median Filtering, Thinning or Skeletonization, Morphological operations such as dilation, erosion, opening, closing and Hit-and-Miss Transform, line detection by Hough Transform, Edge detection, Line following and joining, Region filling, computing Euler Number, Compactness Number and Connectivity Numbers, and the like. The detailed methodology followed in this phase is detailed in Chapter 4.

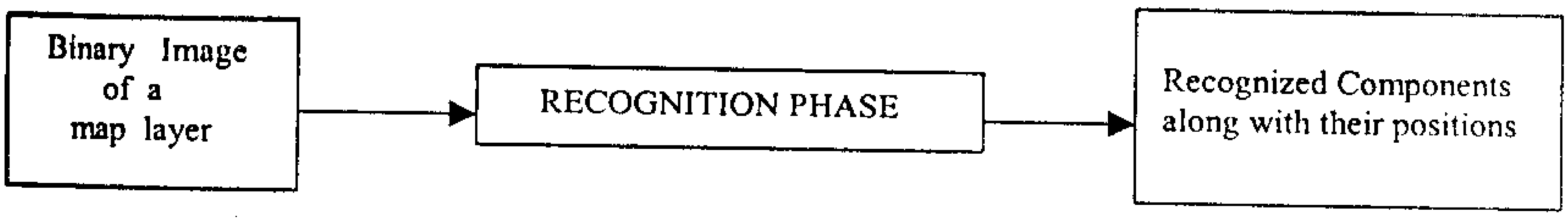


Figure 2.3 – Schematic view of the Recognition Phase.

2.4 Generation of E-Map

For the results of the recognition phase to be practically useful, for example, as input to GIS, a map similar to the original map but recognized by the computer, must be created. This is the e-map or recognized image, which is an image of size equal to the original image, with the names of the features written in place of their respective positions. Such a map requires the integration of the information acquired from the different map layers. A sample e-map is shown in Figure 2.4.

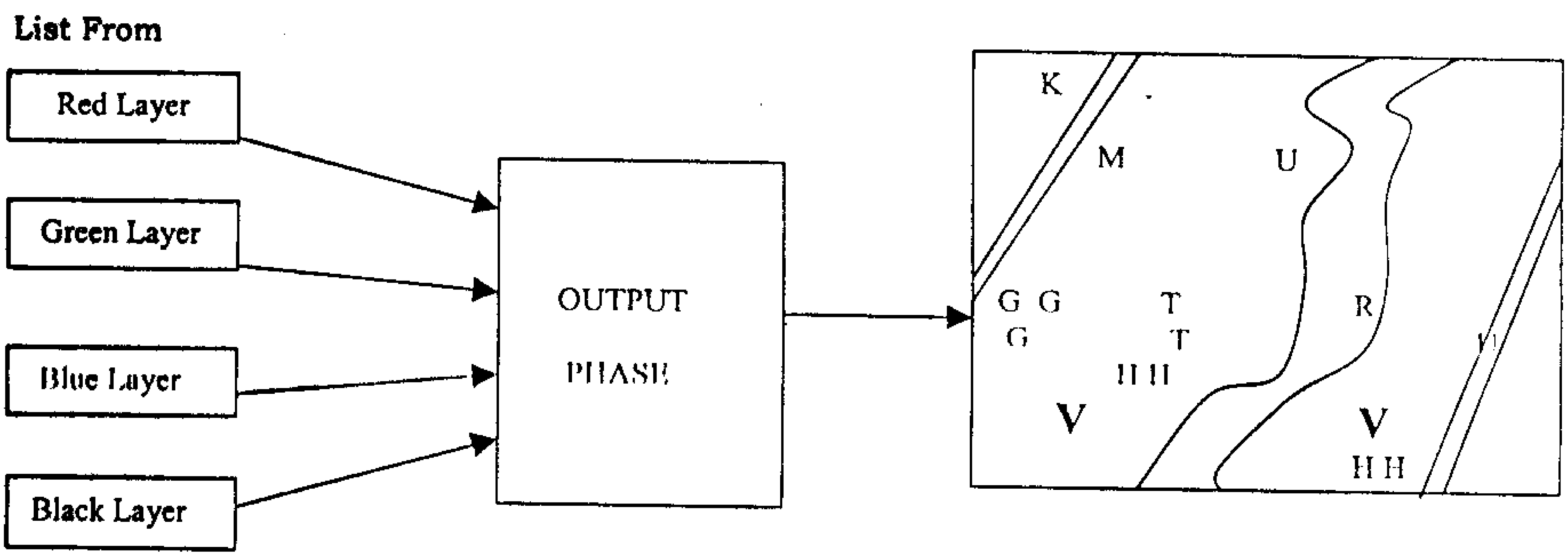


Figure 2.4 – Schematic view of the E-Map Generation Phase giving the Recognized Image.

The symbols used in the final recognized image are :-

Extracted from Red Layer : M – Metalled Road, U – Unmetalled Road, C – Cart Road, H – Huts, V – Village, Town or Human Habitation.

Extracted from Green Layer : T – Tree, G – Grass, F – Green Field or Forest.

Extracted from Blue Layer : R – River, K – Tank.

Extracted from Black Layer : Text strings.

In addition : X – Could not recognize.

✪✪✪✪✪✪✪✪✪✪✪✪✪✪

Chapter 3

LAYER SEPARATION

3.1 Introduction

As already mentioned in the previous chapter, the second phase of the work consists of decomposing the map into four constituent layers, each being a binary image corresponding to the red, green, blue and black regions of the original image.

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 gave poor results due to uneven colour intensity. Studying a number of scanned-in maps, the following problems have been identified :-

(i) The pixel intensities of red, green and blue regions varies widely. In particular, green regions are of low intensity, so that the green component (G) value of the green region is lower than the G value of the red region.

(ii) When the area of objects in a particular layer is quite small or large in comparison to those in the other layers, the cluster corresponding to the dominant layer tends to draw in points from other layers. For example, in the training image, the area of the green regions is much small compared to the red or blue regions. This, coupled with the above fact, tends to draw away pixels that should be in the green layer, to those in red or blue layers.

(iii) The letterings and other features in black had pixel intensity values of R, G, B close to 100, instead of the ideal value 0. Hence, many such pixels tended to be wrongly classified.

Because of the above difficulties, a two-pronged strategy is devised for the purpose of separation of the map into desired layers.

3.2 Clustering Of the Training Image

3.2.1 Image Enhancement

The training image is enhanced such that the R, G, B values of the pixels are either 0 or 255. This led to considerable ease of clustering and thresholding the clusters to binary images at a later stage.

The heuristic followed is as follows. If the R, G, B values of a particular pixel are more or less equal, i.e., within a certain percentage of each other, the colour is essentially gray, and such points are set to either fully black (0,0,0) or fully white (255,255,255) depending upon the average intensity value. (This percentage is fixed after some experimentation.) Otherwise the maximum of the three values is set to 255, and others to zero. If two values are equal and the maximum, both of them are set to 255.

The original training image is shown in Figure 3.1 and the enhanced image in Figure 3.2.

3.2.2 Clustering

The basic objective of any clustering technique is to divide the data points of the feature space into a number of groups or classes so that a predefined criterion is satisfied. Here, the required criterion is intra - class distance. The algorithm used is the **K - Means Algorithm**, where K, the number of clusters, is known apriori. Here K = 5, and the cluster centres chosen are

$C_0 = \{0, 0, 0\}$ for black
$C_1 = \{255, 0, 0\}$ for red
$C_2 = \{0, 255, 0\}$ for green
$C_3 = \{0, 0, 255\}$ for blue
$C_4 = \{255, 255, 255\}$ for white

The feature vector $\mathbf{f} = \{R, G, B\}$
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 $\{\mathbf{f}_i\}$ has been assigned to cluster C_j.

3.2.3 Determination of actual Cluster Centres

The cluster centres obtained by the above process are not the actual cluster centres, but those of the enhanced image. To get the actual cluster centres corresponding to the original image, the average of feature vectors of only those points belonging to a particular cluster (as determined from the binary images output from step 3.2.2) are computed. The feature vector is selected to be the normalized R, G, B values, along with the normalized intensity. That is, for cluster j (j = 0, 1, 2, 3, 4), we have, $\mathbf{C}_j = \{r_j, g_j, b_j, I_j\}$, where,

$$r_j = \frac{1}{nj}\sum_{i \in Cj}\frac{Ri}{Ri+Gi+Bi}, \qquad g_j = \frac{1}{nj}\sum_{i \in Cj}\frac{Gi}{Ri+Gi+Bi},$$

$$b_j = \frac{1}{nj}\sum_{i \in Cj}\frac{Bi}{Ri+Gi+Bi}, \qquad I_j = \frac{1}{nj}\sum_{i \in Cj}\frac{Ri+Gi+Bi}{3 \times 255}.$$

Here n_j is the total number of points in cluster j (Binary image j) and R_i, G_i, B_i are the red, green and blue gray level values of pixel i belonging to the j^{th} cluster.

The feature vector could have been chosen as $\{r_i, g_i, b_i\}$ as before. However this created the problem of differentiating between pixels belonging to black and white classes, as their normalized $\{r_i, g_i, b_i\}$ values tended to be similar. A fourth feature vector element, the intensity I removed this difficulty as black pixels had much lower intensity compared to white pixels.

3.3 Final Separation

Once the actual cluster centres had been determined, the map image is separated into the desired layers by a procedure similar to the **nearest prototype classification algorithm** as detailed below.

Algorithm

Suppose the cluster centres are stored in arrays C_r, C_b, C_g and C_I.

```
for all pixels in the image {
        for i = 0 to 4 {
                Compute normalized {r, g, b, I} vaues for current pixel;
                Determine k = min { (r - Cr[i])² + (g - Cg[i])² + (b - Cb[i])²}
                               i
                and assign the pixel to class k;
                if k = 0 or 4 (black or white classes)
                     Determine min { (r - Cr[i])² + (g - Cg[i])² + (b - Cb[i])²+ (I - CI[i])²}
                                k
                and assign the pixel to that class;
        }
}
```

The pixels assigned to different classes make up binary images corresponding to the different layers. Thus after this step, the training image is completely and (almost) correctly classified into its constituent black, red, green, blue and white layers. These separated layers for the training image are shown in Figures 3.3 to 3.6. Note that for any other image from the same map (and also for images of other maps), this is the first and only step required for separation into layers, since the cluster centres obtained previously are constant for all such images.

✩✩✩✩✩✩✩✩✩✩✩✩✩

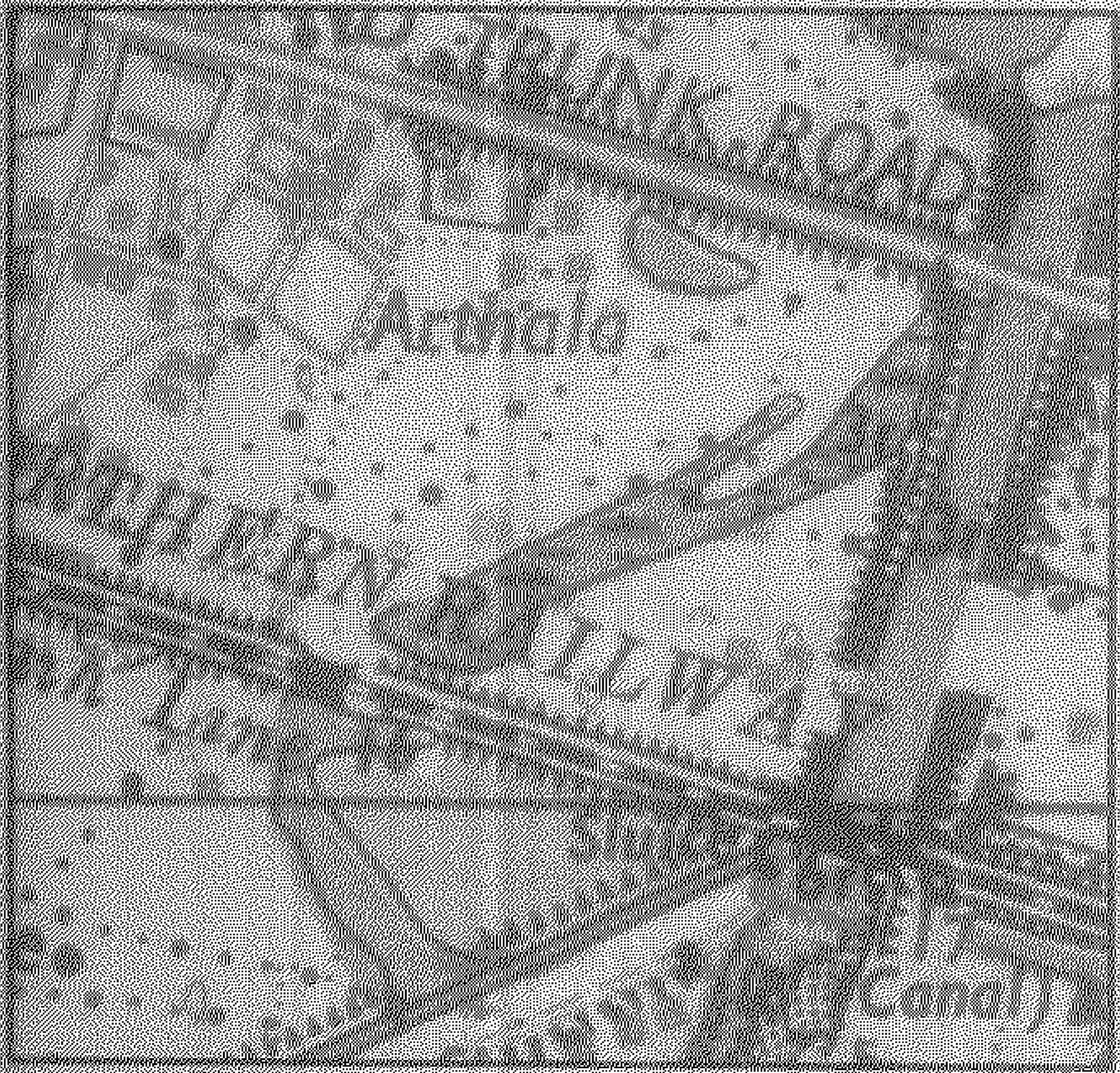


Fig 3.1 – Original Training Image

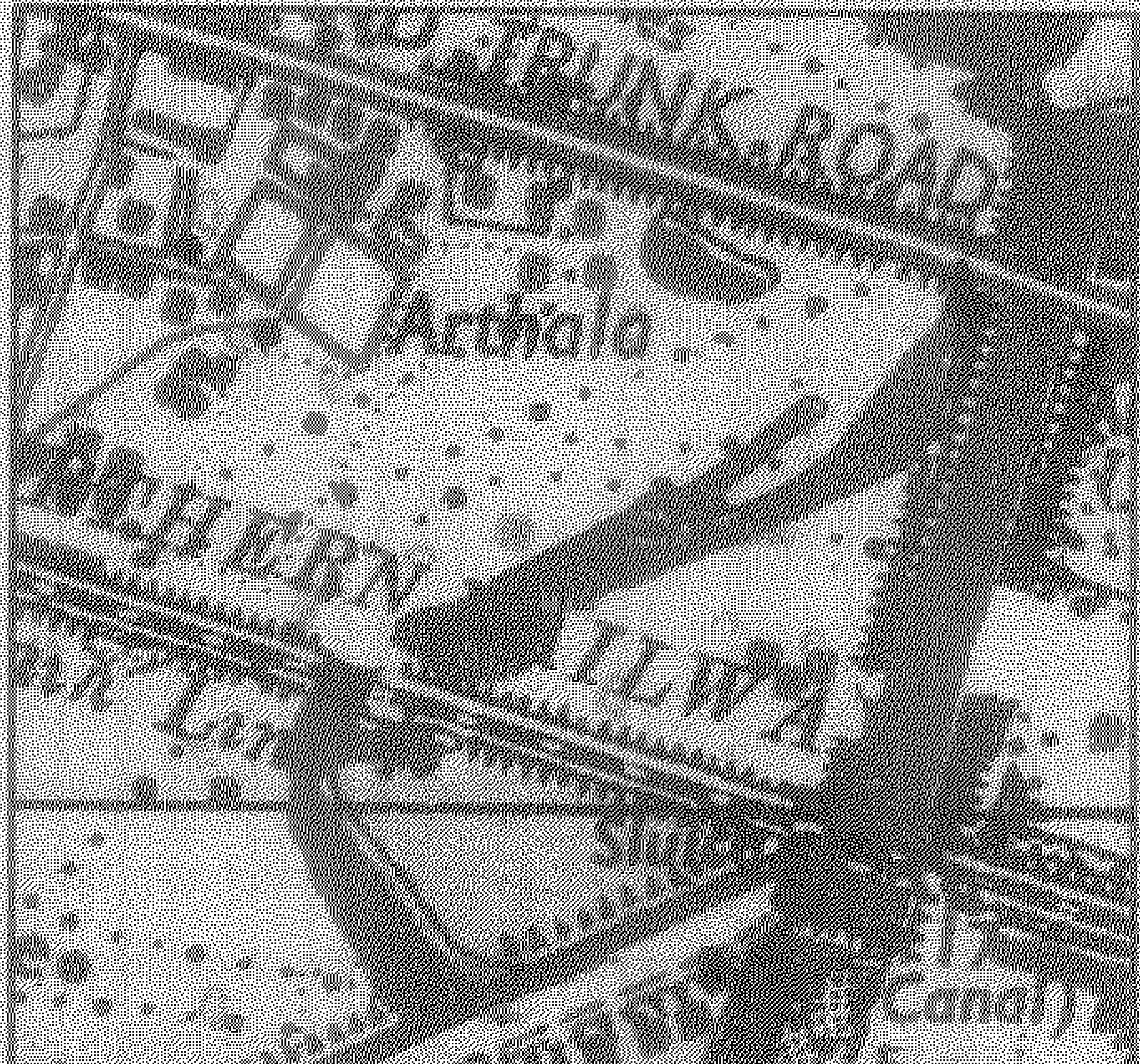


Fig 3.2 – Image after Enhancement

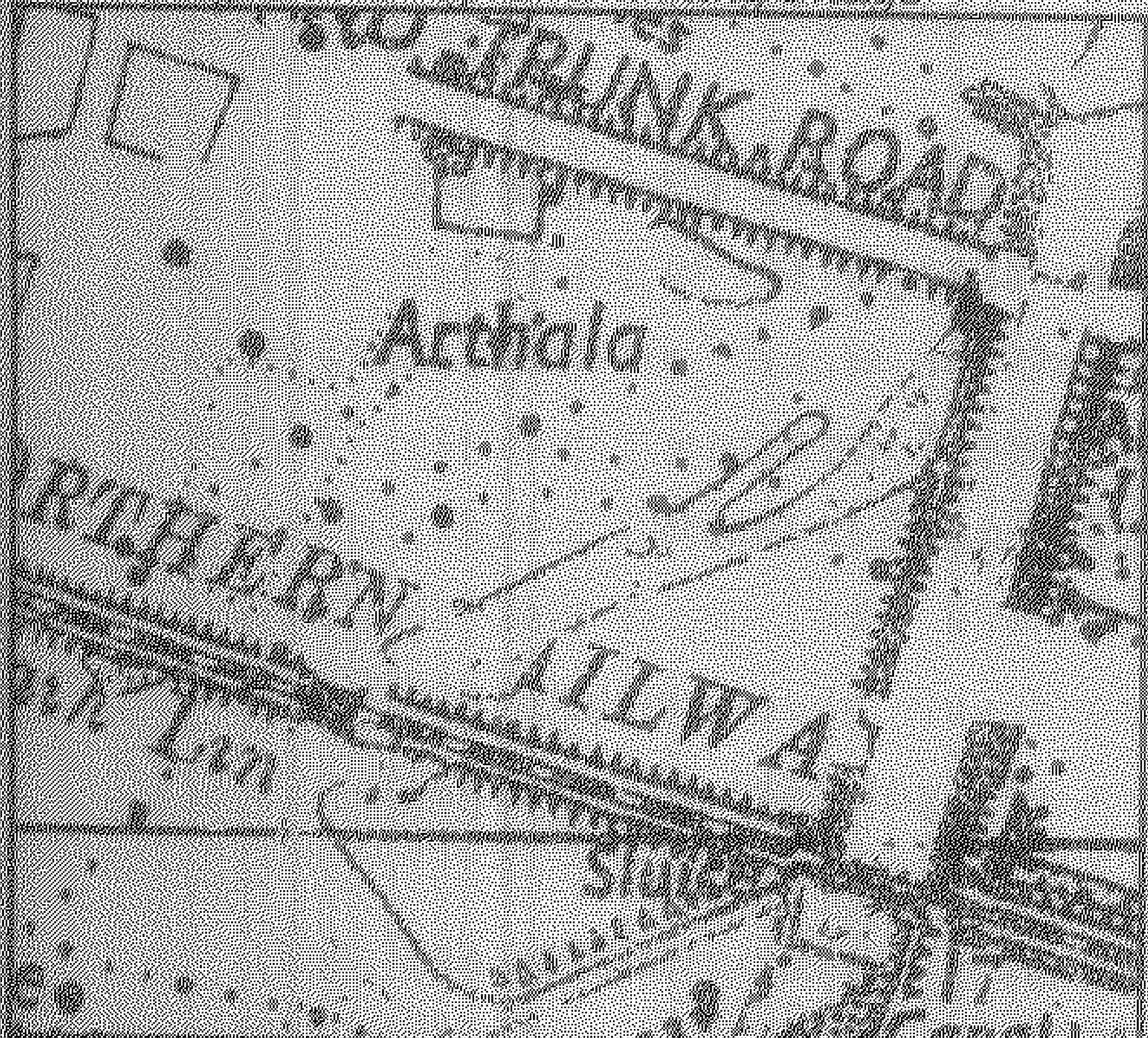


Fig 3.3 – Black Layer of Training Image

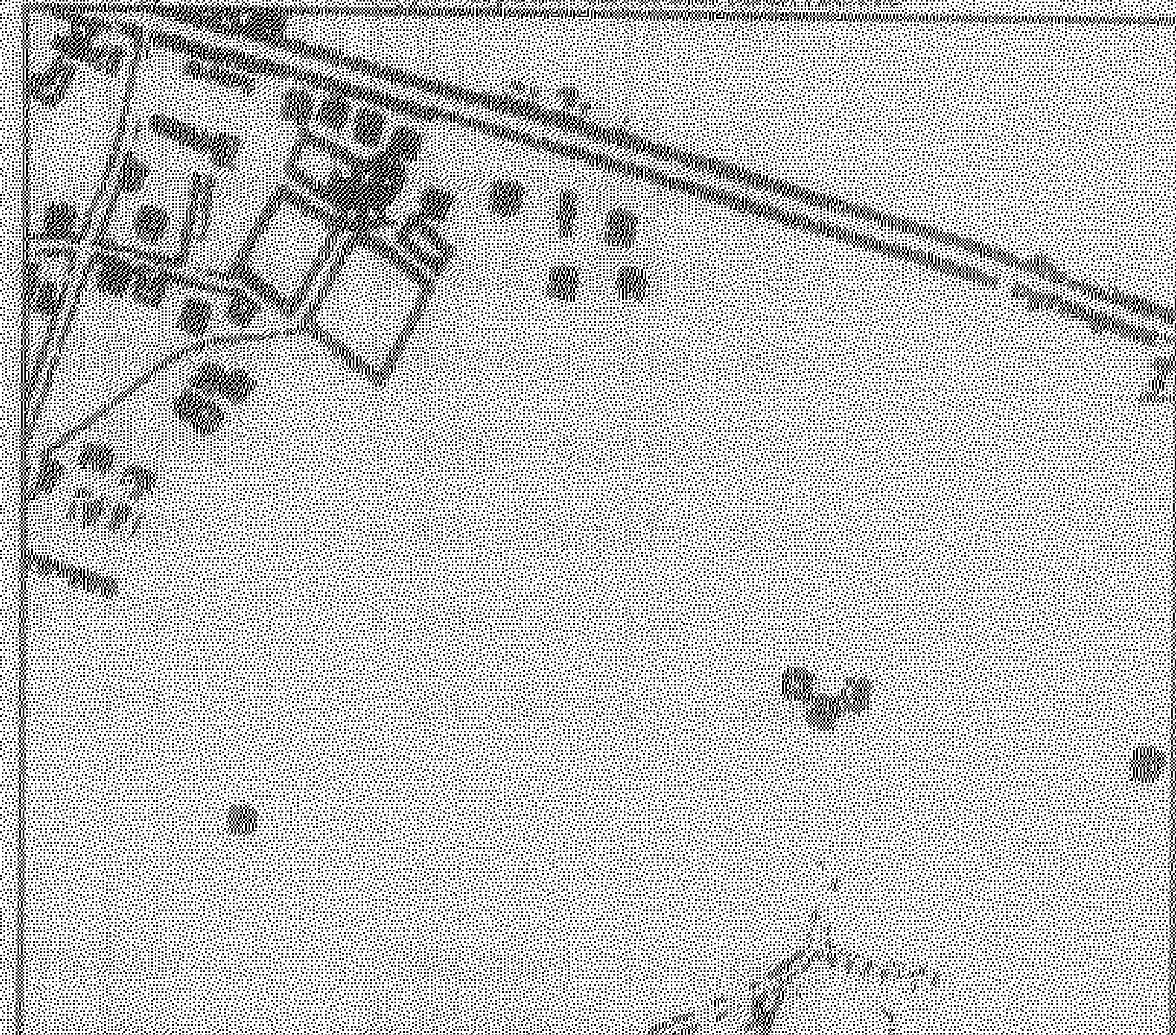


Fig 3.4 – Red Layer of Training Image

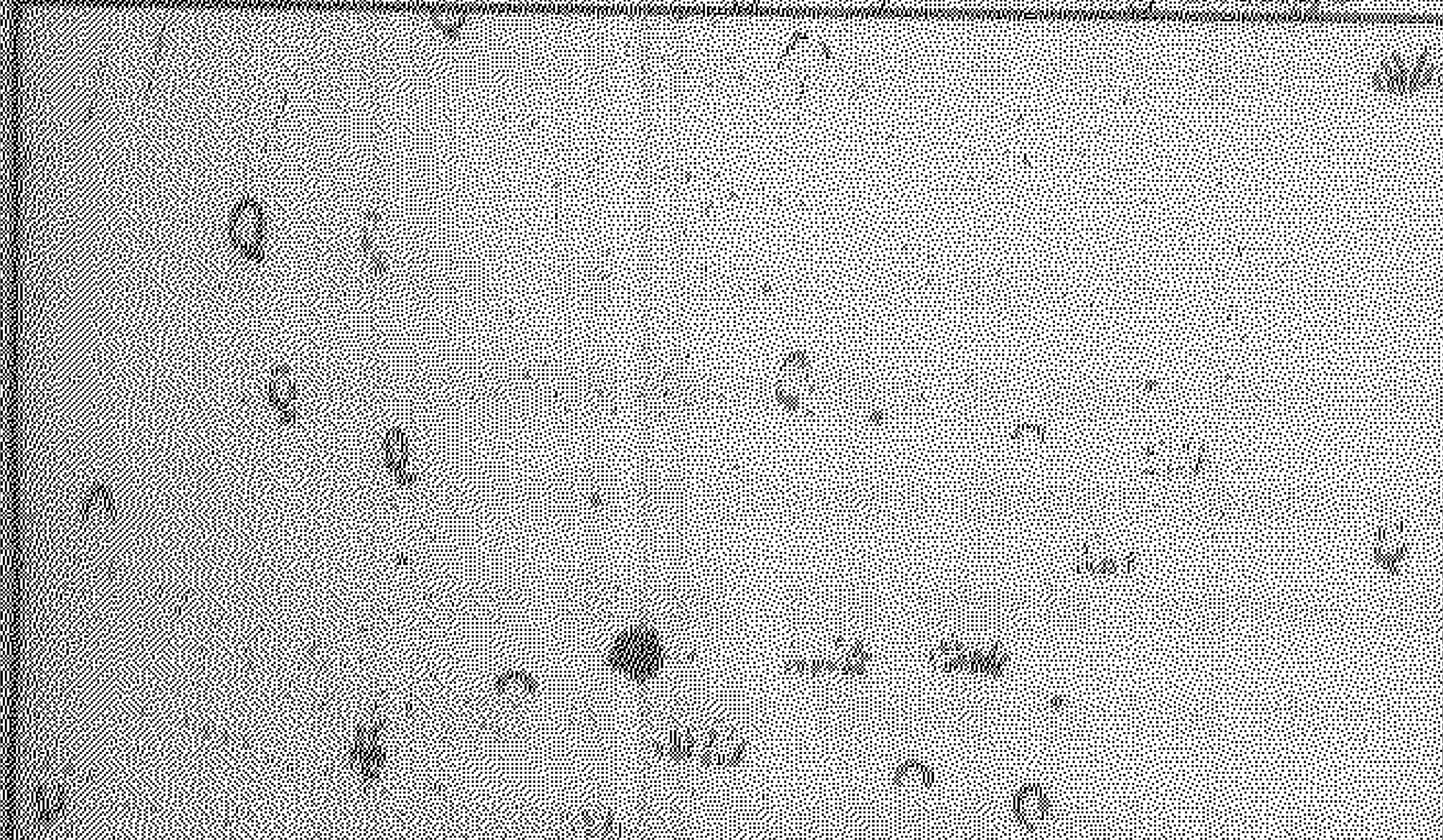


Fig 3.5 – Green Layer of Training Image

Fig 3.6 – Blue Layer of Training Image

Chapter 4

RECOGNITION

4.1 Problems with Recognition

The output of the layer separation phase described in the previous chapter gives four different useful map layers or segments that contains features in black, red, green and blue respectively in the original map. Although this separation is supposed to ease the problem of recognition of symbols from different layers, the problem remains sufficiently complex to yield a ready solution.

It seems that if the legends of the map are scanned, then the direct method of template matching of the legends with the map symbols would achieve the desired recognition. However, this is not the case, as different instances of the same symbol vary in scale and orientation. Statistical Pattern Recognition with features invariant to scale and rotation (Weighted Bounded Several Nearest Neighbour Classifiers) as used by Samet and Soffer in their MAGELLAN and MARCO systems, came next into consideration. Unfortunately, this methodology does not give satisfactory results in this context because of the following reasons.

1. High level of obstruction of geographical symbols due to the map making process – The symbols in different layers of the map intersect with each other, making the problem of recognition very complex.

2. When the map image is classified into the component layers, pixels in the intersections of two or more features is put into one layer, thus making the symbols in the other classes discontinuous or broken. Thus when a bridge (black) runs over a river (blue), the pixels representing the bridge are classified in the black layer, leaving a wide break in the river. Similarly, most of the 'other trees', which are supposed to be closed ovals in green, are found 'cut out' by other features.

3. To alleviate the above problem, researchers have used images of separate map layers as inputs to their system. However, such separate map layers are not available to us.

4. The quality of the print, age-worn paper and other kinds of noise in images makes recognition even more tough.

A great deal of effort thus went into experimenting with heuristic methods with the aim of coming up with robust recognition criteria which would perform reasonable well even in the presence of the above mentioned difficulties.

Before a description of the methods of specific feature recognition starts, a mention of Component Labelling is in order, as it is used with the images of separated layers, though in different order and different objectives.

4.2 Component Labelling and Noise Removal

The two-pass algorithm [Rosenfeld & Kak] is used. The method starts with an empty equivalence table.

Pass I : Binary image is scanned in raster order, from top to bottom, left to right. Once a 1-pixel is found, its top lest neighbours are checked. If all of them are 0-pixels, a new label is assigned to the candidate pixel. If a unique label exists in this neighbourhood, that label is assigned to the candidate pixel. If more than one label exists in this neighbourhood, any one of them is assigned to the candidate and a note of the equivalence of the labels is entered in the equivalence table.

Pass II : In consultation with the equivalence table, all equivalent labels are labelled by a unique label.

For noise removal, the area of each component is counted, and components with very small area (below a user specified threshold) are removed from the binary image.

4.3 Recognition of Features in the Green Layer

According to the legends of the map (shown in Figure 4.1), the objects in the green domain included grass, scrub, palm trees, plantain, conifer, bamboo and 'other trees'. Extensive traces of green indicate green fields and forests. But since plantains conifers and bamboos have not been found in the training or test images, it has been decided to look for palm trees and 'other trees' and label them as just 'trees'.

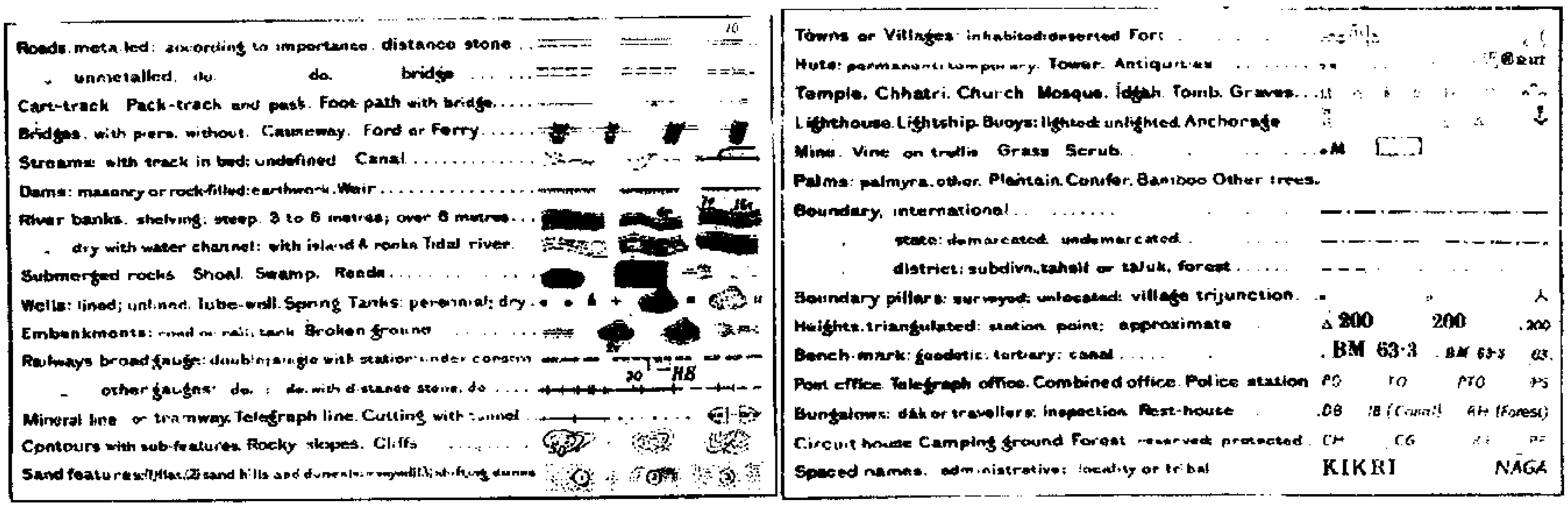


Figure 4.1 – Sample legend of Survey Map

The specific steps performed are discussed below.

Step 1 - Removing Negative Noise : 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 break in strokes. This type of noise may be removed by morphological closing operation.

If A and B be subsets of a rectangular domain D, then A closed by B is defined as

$$A \bullet B = (A \oplus B) \ominus B$$

where $\oplus$ and $\ominus$ denote dilation and erosion respectively.

A dilated by B is defined as

$$A \oplus B = \{ p \in D \mid p = a + b \text{ for some } a \in A, b \in B \}$$

A eroded by B is defined as

$$A \ominus B = \{ p \in D \mid p + b \in A \text{ for every } b \in B \}$$

Closing tends to smooth sections of contours, fuse narrow breaks and long thin gulfs, eliminate small holes and fill gaps in the contour. Here it is expected to fuse the arms of grasses and shrubs and fill the vacant areas within green fields, such that they would be counted as single components. However, the success of the operation depends upon the choice of the structuring element B. The selection of the structuring which gives best results (i.e. joins most of the desired gaps without joining separate components) has to be done by visual inspection.

Step 2 – Removing Positive Noise : In a binary image, if noise converts a white pixel to black pixel, it is considered as positive noise. This type of noise is cleaned by component - counting and removing small sized components. The reason why this step is performed after the closing operation is that otherwise the blades of grass as well as major portions of green fields (which are not connected components) would have been eliminated as noise.

The image obtained from the green layer of the training map (Figure 3.5) after performing the closing and component labelling operations is shown in Figure 4.2.

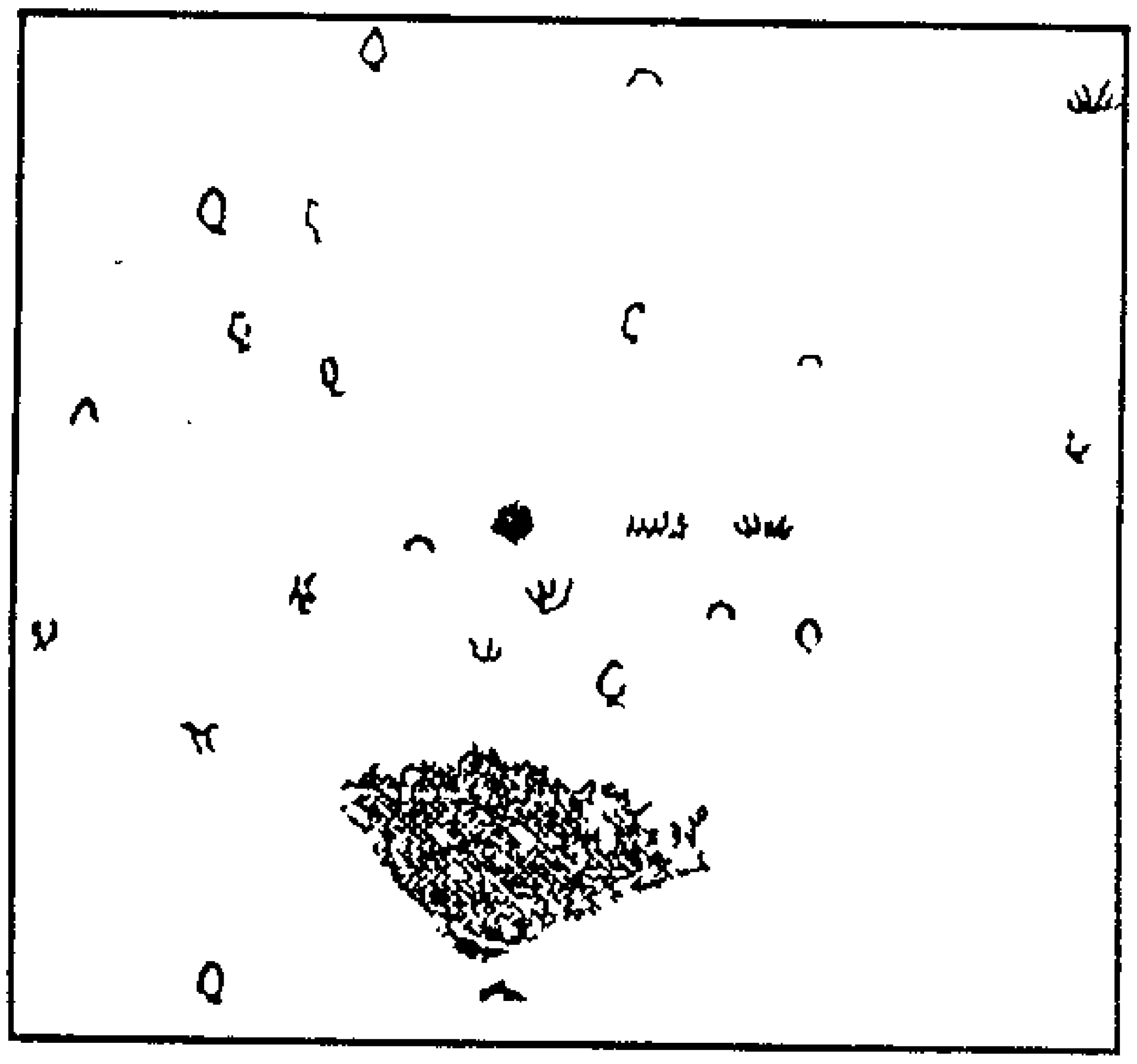


Figure 4.2 – Image of Figure 3.5 after removal of negative and positive noise. ➡

<u>Step 3</u> - Computing Euler Number : The *genus* or *Euler Number* of an image is defined as the Number of Components - Number of Holes.

If the set S represents the entire domain of support and $b_s(r,c)$ be the indicator fuction of the image, then the genus $G(b_s)$ of the image is computed as

$$G(B_S) = \#(S) - \sum_{i=1}^{4} \#(S \ominus K_i) + \sum_{i=5}^{8} \#(S \oplus K_i) - \#(S \ominus K_9)$$

where #(S) means the number of elements in S, and K_i's are the structuring elements as shown in Figure 4.3.

K_1

X	X

K_2

X	
	X

K_3

X	
X	

K_4

_	X
X	

K_5

X	X
X	

K_6

X	
X	X

K_7

_	X
X	X

K_8

X	X
	X

K_9

X	X
X	X

Figure 4.3 – Structuring elements for computing Euler Number

Ideally, if all components with Euler Number 0 are listed, all trees should have been identified as they have exactly one hole. However, because of the obstruction of geographical symbols as mentioned in Section 4.1, many of the trees do not satisfy this criterion. Thus, some additional criterion is required to recognize trees.

Step 4 - **Thinning or Skeletonization :** A closer examination of Figure 4.2 reveals that most of the trees have one Y-shaped junction (with three arms), whereas grasses and green fields have more of them. Thus, the number of such junctions could serve as a criterion for recognition of trees. Such junctions can be identified by locating points which have Connectivity Number 3. However, to correctly identify junction points by Connectivity Number, the components must be thinned. Hence, the fourth step consists of thinning or skeletonization of the components.

Figure 4.4(a) displays a component that is a tree present in Figure 4.2 and Figure 4.4(b) its thinned version. The pixels representing the object are shown as '1', whereas those in the background are shown as dots.

```
. . . . . . . . . .
. . . . 1 1 1 . . .
. . . . 1 1 1 1 . .
. . . 1 1 1 1 1 . .
. . 1 1 1 . 1 1 . .
. . 1 1 . . 1 1 . .
. . 1 1 . . . 1 . .
. . 1 1 . . 1 1 . .
. . 1 1 1 1 1 . . .
. . . 1 1 1 1 . . .
. . . . 1 1 1 . . .
. . . . 1 1 1 1 . .
. . . . . . . . . .
```
(a)

```
. . . . . . . . .
. . . . 1 . . . .
. . . 1 . 1 . . .
. . 1 . . . 1 . .
. . 1 . . . 1 . .
. . 1 . . . 1 . .
. . 1 . . . 1 . .
. . 1 . . 1 . . .
. . . 1 1 . . . .
. . . . 1 . . . .
. . . . . 1 1 . .
. . . . . . . . .
. . . . . . . . .
```
(b)

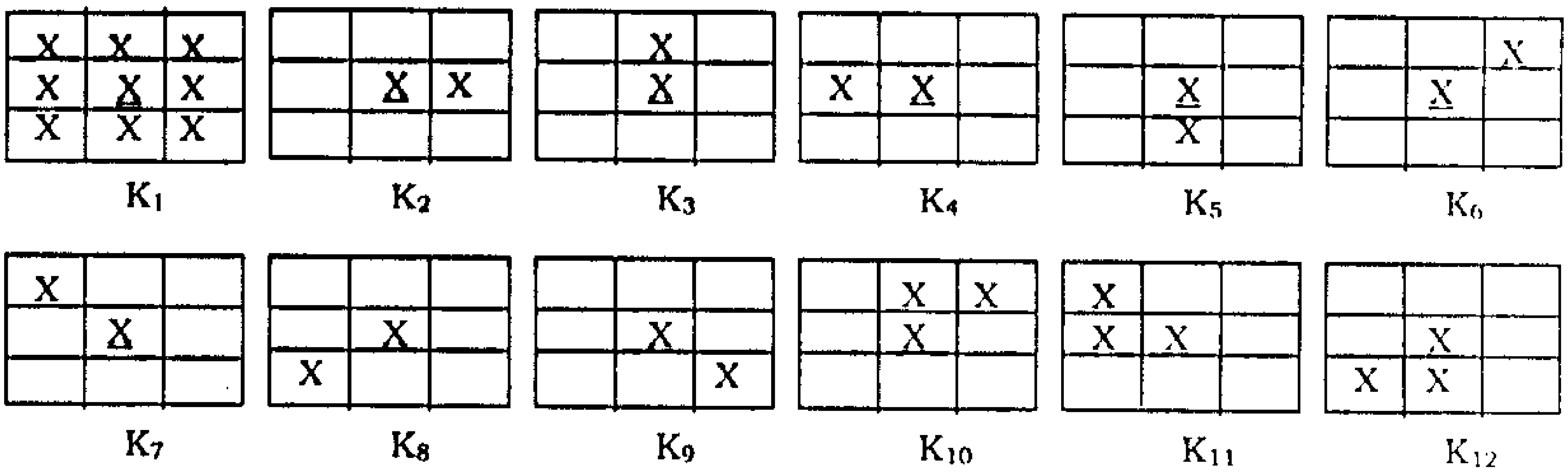
```
. . . . . . . . .
. . . . 2 . . . .
. . . 2 . 2 . . .
. . 2 . . . 2 . .
. . 2 . . . 2 . .
. . 2 . . . 2 . .
. . 2 . . . 2 . .
. . 2 . . 2 . . .
. . . 2 3 . . . .
. . . . 2 . . . .
. . . . . 2 1 . .
. . . . . . . . .
. . . . . . . . .
```
(c)

Figure 4.4 – (a) A component of 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.

Step 5 - **Computing Rutovitz Connectivity Number :** Also known as the 'crossing number', this is defined as the number of transitions from the object to its background if we travel around the 8-neighbourhood of a candidate pixel belonging to the boundary of an object. The connectivity number at pixel *(r,c)* of an object is computed using

$$C(r,c) = \max\{ \sum_{i=2}^{9} (b_s \ominus K_i)(r,c) - \sum_{i=10}^{17} (b_s \ominus K_i)(r,c)\, ,\ 5(b_s \ominus K_i)(r,c) \}$$

where the structuring elements K_i are shown in Figure 4.5.



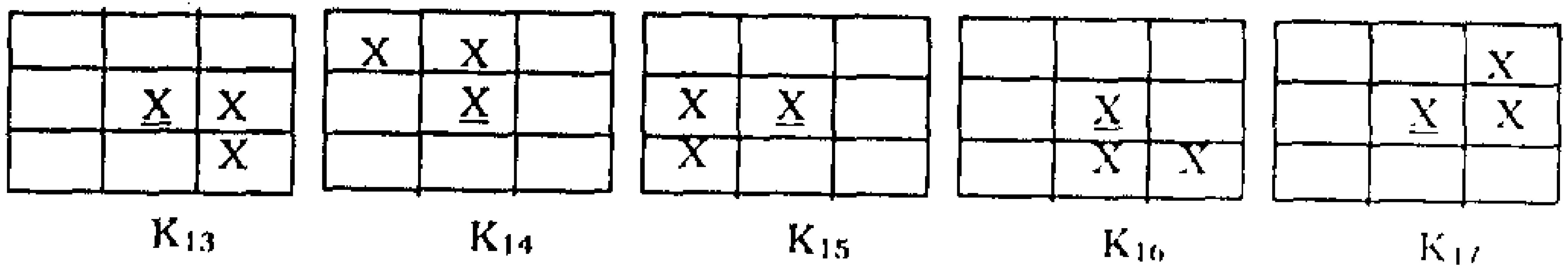


Figure 4.5 – Structuring elements for Connectivity Number Labelling.

Step 6 – Decision : Components with exactly one hole (Euler Number = 1) or one point with Connectivity Number = 3 identify many of the trees, but many trees have been so cut that they do not have even a single Y junction. To identify such components as trees, the maximum (*MAX*) number of 1-0 transitions (object - background transitions) in a single row of each component is determined. The final rules for the green layer are :

- If the component has Euler Number = 1 and *MAX* = 2, then it is a tree.
- If *MAX* = 2 and Rutovitz Connectivity Number <= 2, then the component is a tree.
- If *MAX* lies between 3 and 6, the component is a grass.
- If *MAX* is greater than 6, the component is a green field or forest.

Some examples of various types of objects in the green layer are shown in Figure 4.6.

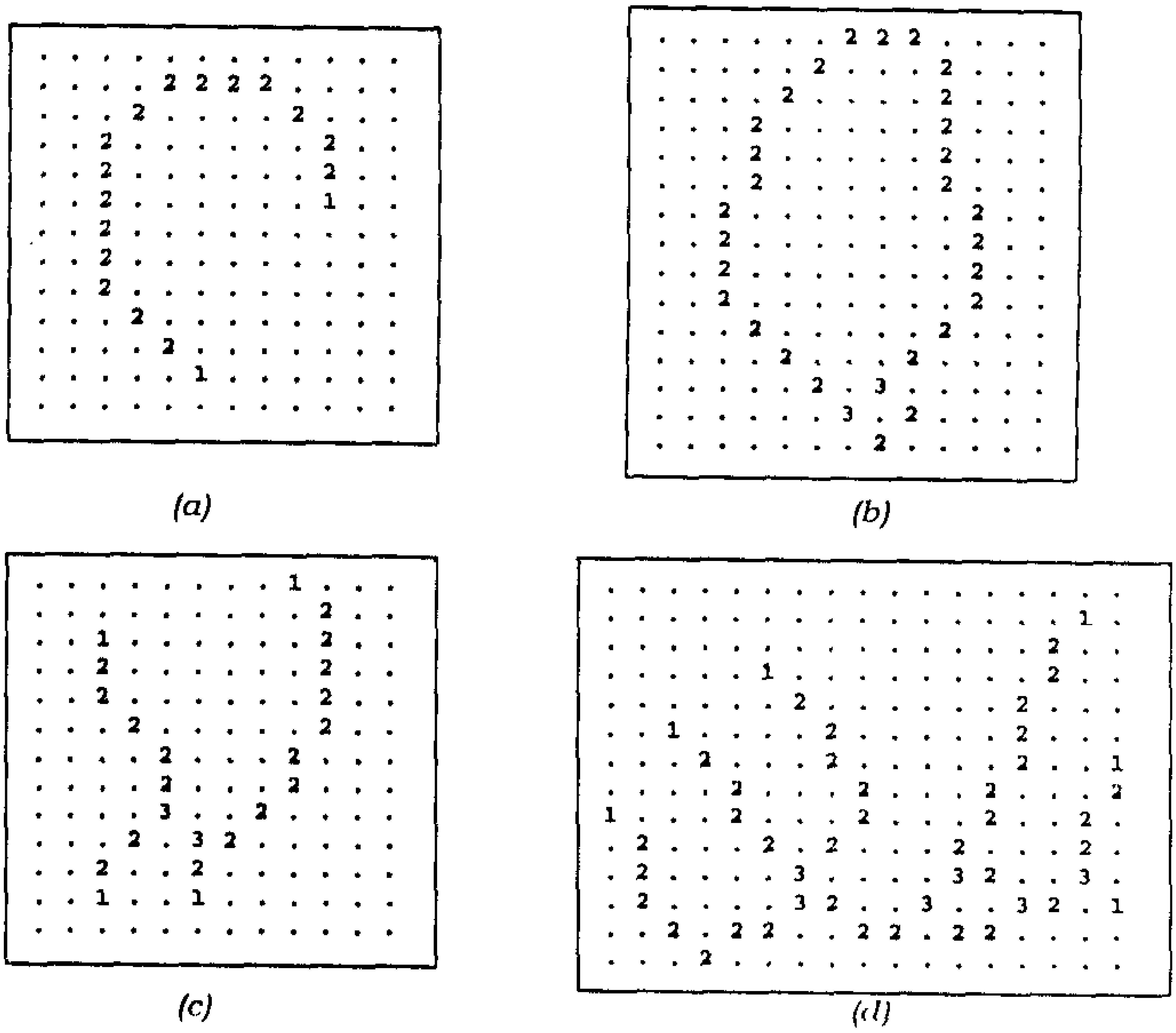


Figure 4.6 – (a) Example of a tree that has no hole and no point with Connectivity No. 3. (b) Example of a tree that has one hole and two points with Connectivity No. 3. (c) Example of a tree that has no hole but one point with Connectivity No. 3. (d) Grass.

The final output of Figure 4.2 after recognition and generation of E-map of the green layer is shown in Figure 4.7. Note that the minimum bounding octagon for the field has been drawn in the E-map.

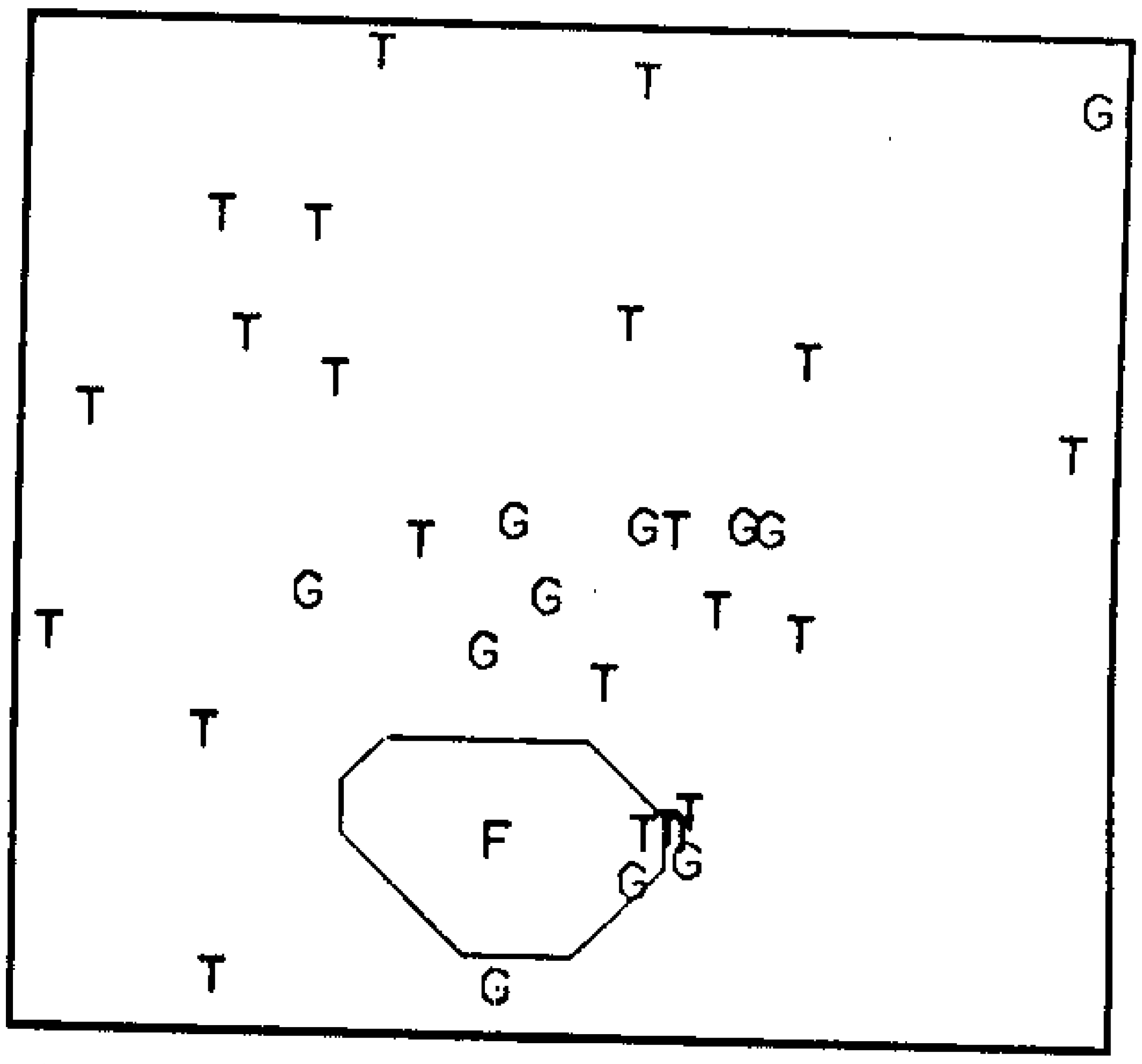


Figure 4.7 – Recognized map for the green layer of the training image (cf. Figure 4.2). 'T' represents tree, 'G' grass, 'F' green field, and 'X' represents unrecognized objects.

4.4 Recognition of Features in the Blue Layer

The few features available in blue can be grouped under two broad categories – river and tank (see the map legend, Figure 4.1). The greatest problem is with identifying rivers, since they are intersected by bridges and dams, and consequently have large parts missing from the blue layer, as evident from Figure 3.6. There is only one river in that figure; however, it is divided into many components. These components 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. Once such components are joined, recognizing the features become easier. The specific steps of recognition are stated below.

Step 1 – Component Joining and Noise Removal : The user is expected to input the components to be joined. The end points of such components are joined by straight lines and the enclosing area is filled using *Flood-Fill* algorithm. This is followed by a Component Labelling step that removes postive noise as specified in section 4.2. Negative noise is removed by ***Median Filtering***.

The result of this step on the blue layer of the training image (Figure 3.6) is shown in Figure 4.8.

Step 2 – **Recognition** : The decision criteria is as follows :-

- If an object touches three or more sides of the image, it is a river.
- If an object touches two *non-adjacent* sides of the image, it is a river.
- If an object touches two adjacent sides of an image, it is a tank.
- If an object touches one or no side of an image, it is a tank.

Step 3 – **Boundary Extraction and Output** : Although strictly speaking, this falls in the E-map generation phase, it is mentioned here for the sake of completeness. Boundary extraction of the components in the blue layer is performed and this boundary is drawn in the output map. Also, the symbol 'R' or 'K' is printed at the centre of the corresponding object depending on whether it has been identified as river or tank respectively. The resulting image for the training image is shown in Fig 4.9.

Figure 4.8 – Image of Figure 3.6 after Step 1

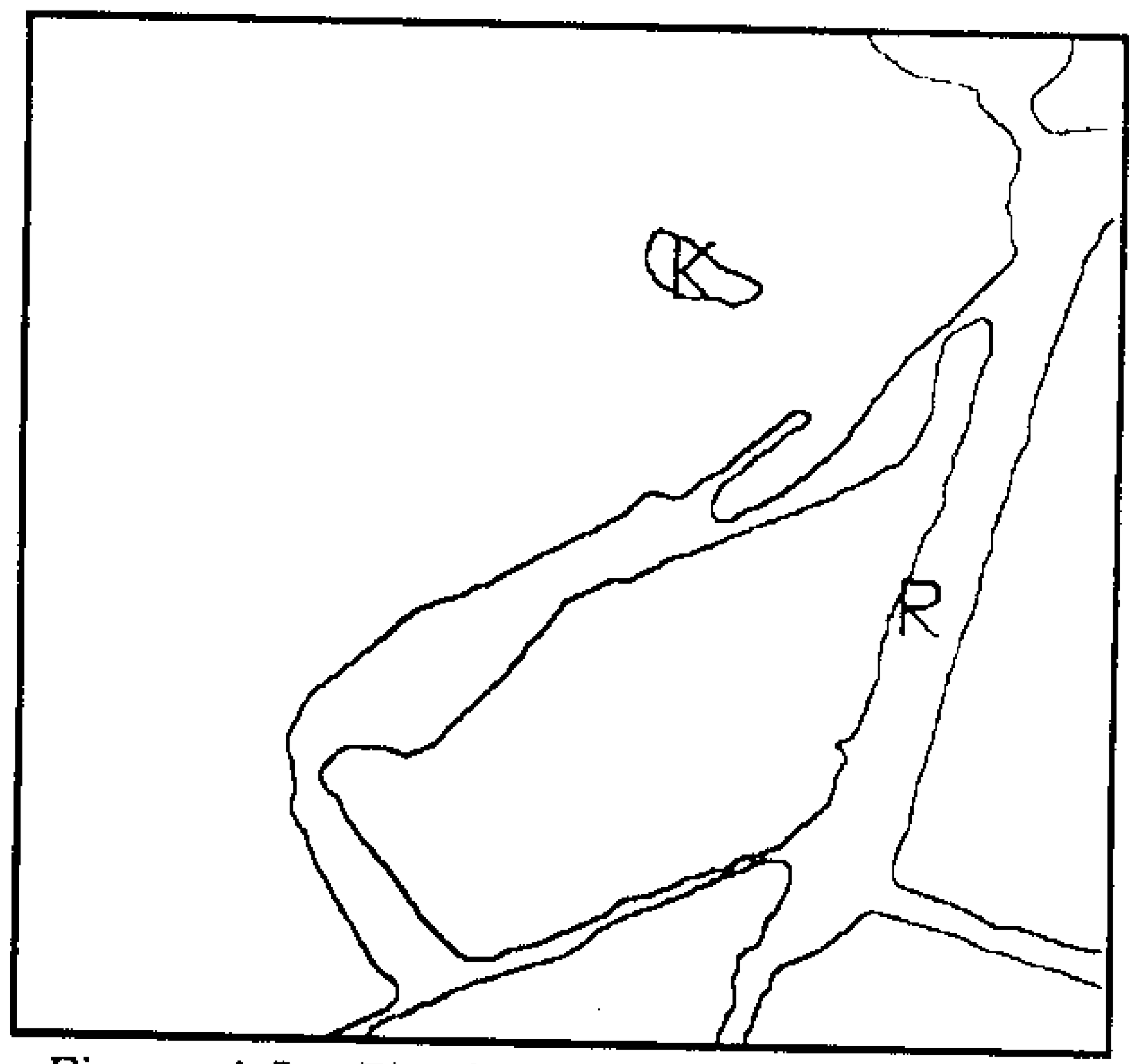


Figure 4.9 – Final e-map for blue layer

4.5 Recognition of Features in the Red Layer

The presence of several different types of features in red (See the map legend, Figure 4.1) and the occurrence of several of them in aggregation (see the red layer in the training image, Figure 3.4) makes the isolation and recognition of these features extremely difficult.

The features extracted from the red layer along with their characteristics that form the basis of their separation is shown below. Also given are the symbols used to represent them in the E-map.

Table 4.1 – Prominent Features found in the Red Layer.

	Feature	Characteristics	Symbol used in E-map
1.	Huts	Small rectangular blobs	H
2.	Metalled Roads	Solid double parallel lines	M
3.	Cart Roads	Solid Single line	C
4.	Unmetalled Roads	Dashed double parallel lines	U
5.	Village / Town / Human Habitation	Aggregation of small and large-sized filled or unfilled rectangles, possibly with roads running in between.	V

The approach taken for recognizing components in the red layer is different from that taken in the previous two layers. Here different types of features are isolated one type at a time, in the order as shown in Table 4.1, based on their geometric and shape properties. The process may be thought of as sieving, and the features isolated at each stage of sieving are removed from the image of red layer and are put into the E-map. The specific steps of implementation are described below.

Step 1 - Line Joining and Noise Removal : Because of noise, many line segments in the red layer are broken. This is especially true for long straight lines representing different types of roads. Thus the first step is to join the broken lines. For this the entire image is thinned and the end points of the lines are identified. (An end point is a point that has only one neighbour in its 8-neighbourhood.) After identifying the end points, they are joined with their nearest end point neighbour, provided that they are within a specified distance of each other. This joins many broken lines in which the breaks are a few pixels wide.

However, there are cases where the above technique does not work. Examples are as follows: -

a) Unmetalled roads are represented by dashed parallel lines. As shown in Figure 4.10, the above algorithm would join end points 1 with 2, and 3 with 4, instead of joining 1 with 3 and 2 with 4.

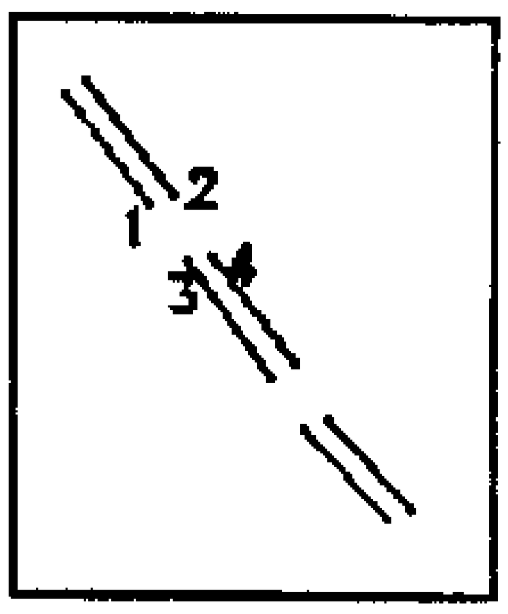


Figure 4.10

b) Cases where the break is quite large. Attempts may be made to join such lines by increasing the threshold, but this introduces the problem of joining undesirable line segments.

To get a somewhat satisfactory solution to these problems, an approach using human intervention and based on directional mathematical morphology is used. The direction of probable lines are first determined using ***Hough Transform*** (explained later in the context of Step 4). The user then selects structuring elements according to the direction of the matching lines. Finally directional morphological closing operations are applied on the original image to join all broken lines satisfactorily. It may be noted that attempts are not made to join dashed lines of the form in Figure 4.10.

After line joining, noise-cleaning operation is done using component labelling and removing the small components as described in Section 4.2. The image resulting from the red layer of the map (Figure 3.4) after these operations is shown in Figure 4.11.

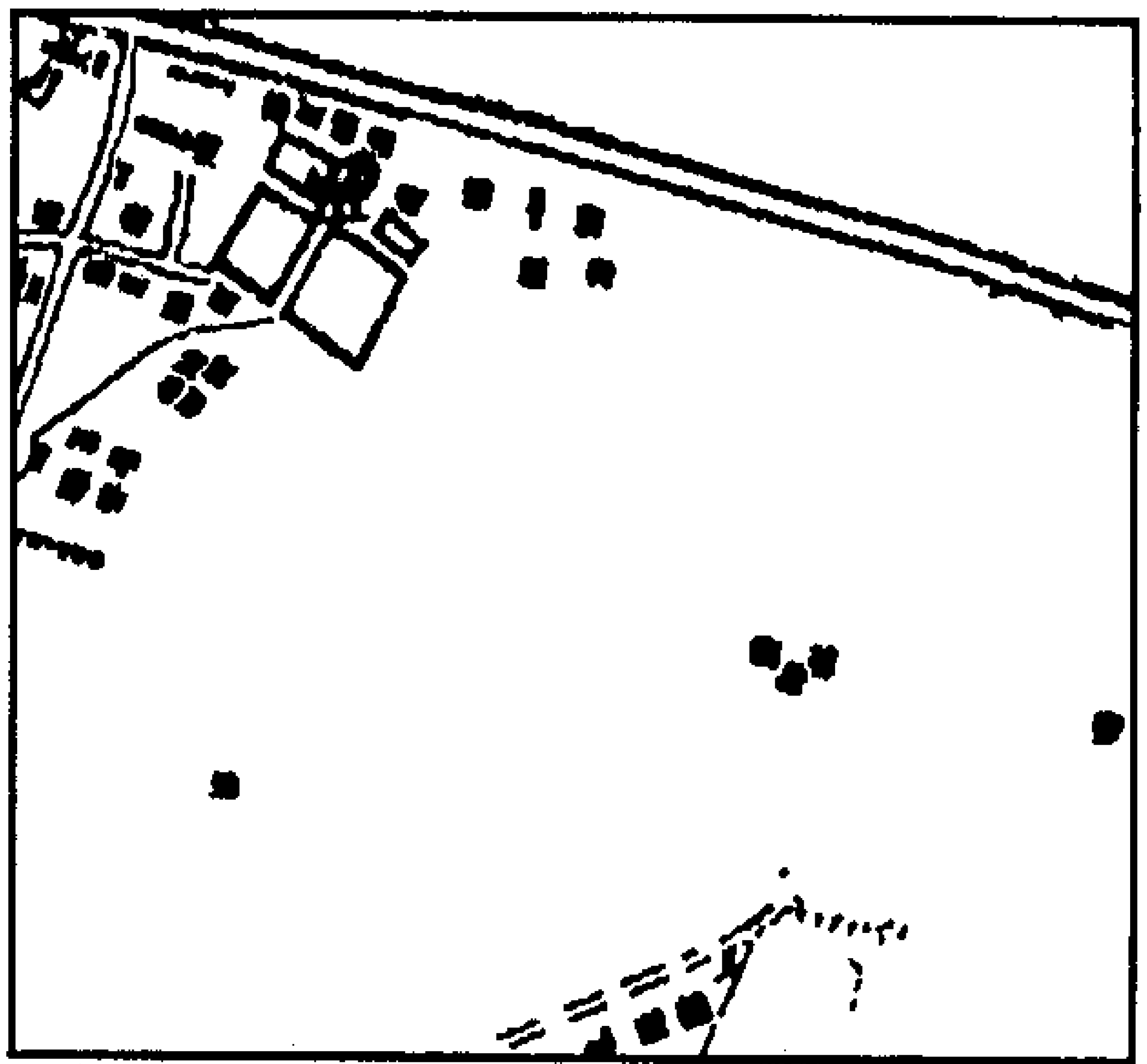


Figure 4.11 – Red Layer of the training Image after line joining and noise-removal.

Figure 4.12 – Huts extracted from Fig 4.11

__Step 2__ – **Isolation of Huts** : The first feature to be extracted from the preprocessed image of the red layer are the huts. These are separated using the shape features *(i)* ***aspect ratio***, the ratio of the smaller dimension to the larger dimension of the bounding box or minimum enclosing rectangle, 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. The components isolated as huts from Figure 4.11 are shown in Figure 4.12.

<u>Step 3</u> - Isolation of Metalled and Cart Roads: The unique geometric characteristic of roads is their high length-to-width ratio. In other words, they have quite a 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 components with a low aspect ratio or a low vacant ratio (less than a threshold, chosen to be 0.15 in this case) are recognized as metalled or cart roads. The components isolated by these criteria for the training image is shown in Figure 4.13.

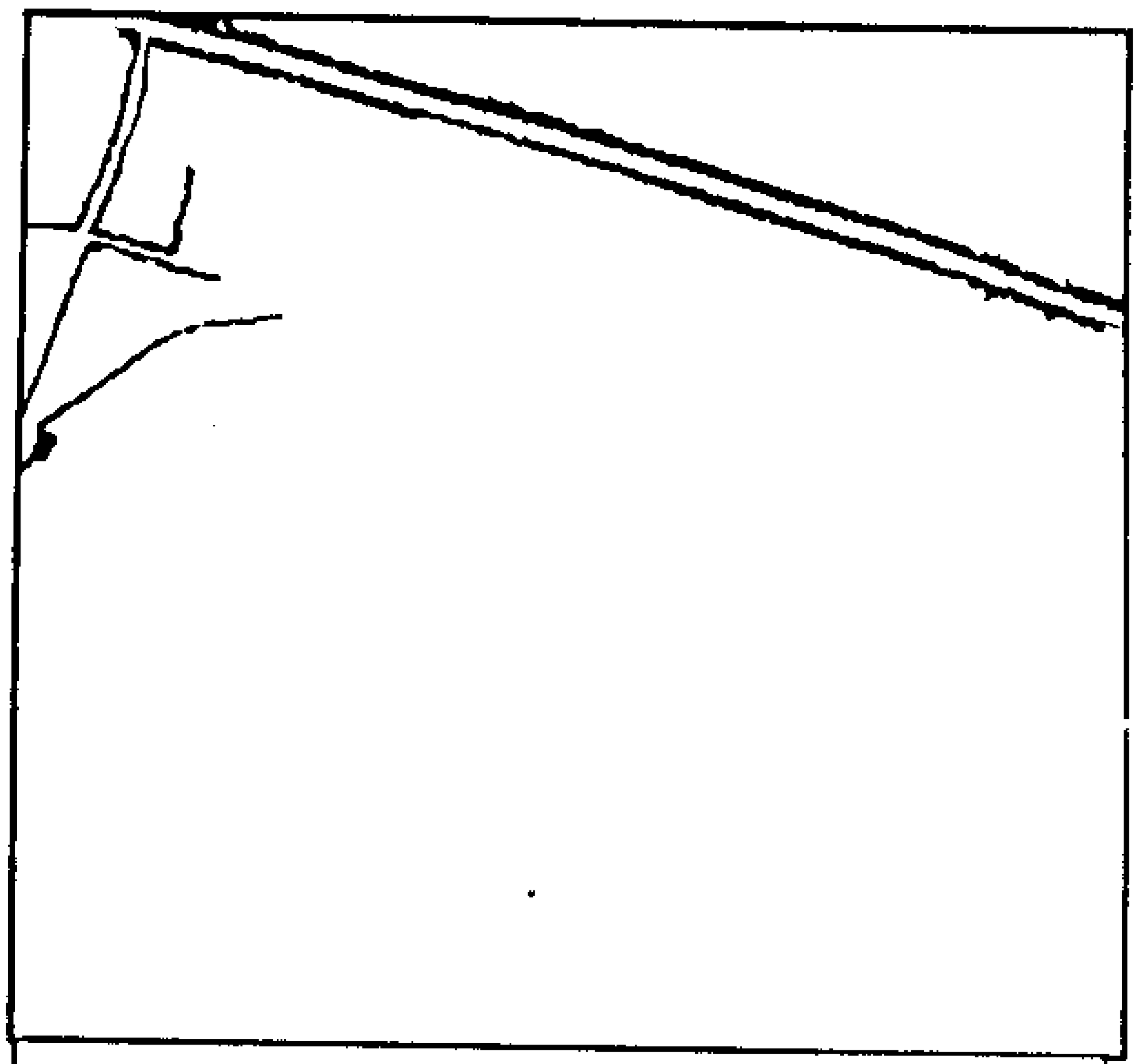


Figure 4.13 – Metalled and Cart Roads isolated for the training image.

<u>Step 4</u> - Distinguishing Metalled Roads from Cart Roads : Once identified, metalled and cart roads are drawn on a separate image and Hough Transform is applied to them. Before proceeding further, it would be appropriate to mention briefly the purpose and methodology of the Hough Transform.

The Hough Transform considers global relationships among pixels to determine whether or not a subset of them lie on a curve of specified shape. Here it has been used to detect straight lines. It is based on the idea that if N points lie on the straight line

$$p = r \cos \theta_l + c \sin \theta_l ,$$

then in the parameter space (p, θ), there are N concurrent lines passing through the point (p_1, θ_1). Thus the (p, θ)-plane is approximated by a two-dimensional ***accumulator array*** A(p, θ). Then, for all θ lying between -180° and 180°, and for all 1-pixels having co-ordinates (r', c'), the value of $p = r' \cos \theta + c' \sin \theta$ is computed, and A(p, θ) is incremented. For each accumulator cell, information regarding r_{min}, r_{max}, c_{min}, c_{max} are also kept based on the points that contribute to that cell. This is done in order to keep track of the extent of the lines. At the end of the computation, the local maxima in A(p, θ) detect the (p, θ) that form lines in the original image.

Once the Hough Transform has been determined, it is easy to determine whether the road is a metalled road or cart road (i.e., double line or single line). If the Hough transform detects two lines having the same inclination but only a few pixels apart (i.e., $\theta_1 = \theta_2$ and $|p_1 - p_2|$ less than a threshold value, here chosen to be 10), then the two lines are parallel lines representing a metalled road. Otherwise, the roads are cart roads. Correspondingly, the lines are drawn on the E-map and labelled as 'M' or 'C'.

It should be mentioned here that the above method assumes that the roads are straight. This could be extended for curved roads if they are approximated as being made up of straight-line segments.

<u>Step 5</u> – Detecting Unmetalled Roads : We now have the image after subtracting the huts, metalled and cart roads. From this image, components having area less than a specified threshold (chosen to be 50 here) are removed and placed in a new image in order to search for the possible presence of unmetalled roads (dashed lines). The components isolated by this criterion from the subtracted training image are shown in Figure 4.14. Here again, Hough Transform is used to determine the presence of two parallel lines. If two such lines are found, these lines are drawn in the output image or E-map using the p, θ and end-point information, and are labelled as 'U'. (The output of this reconstruction is shown in Figure 4.15.) Otherwise, no action is taken because Hough Transform may detect an isolated line from sporadic points, and such a line should be neglected.

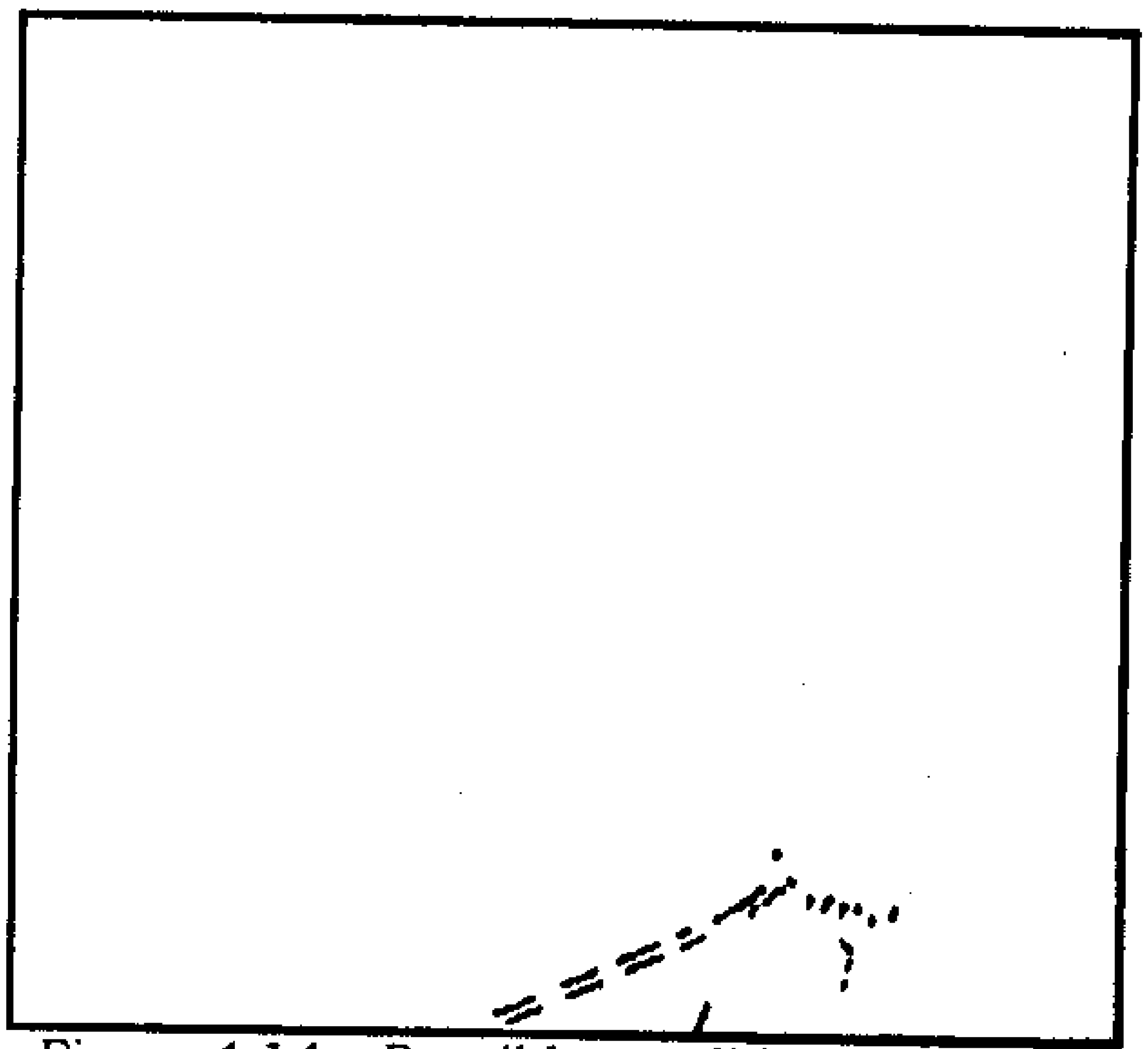


Figure 4.14 – Possible candidates for unmetalled roads.

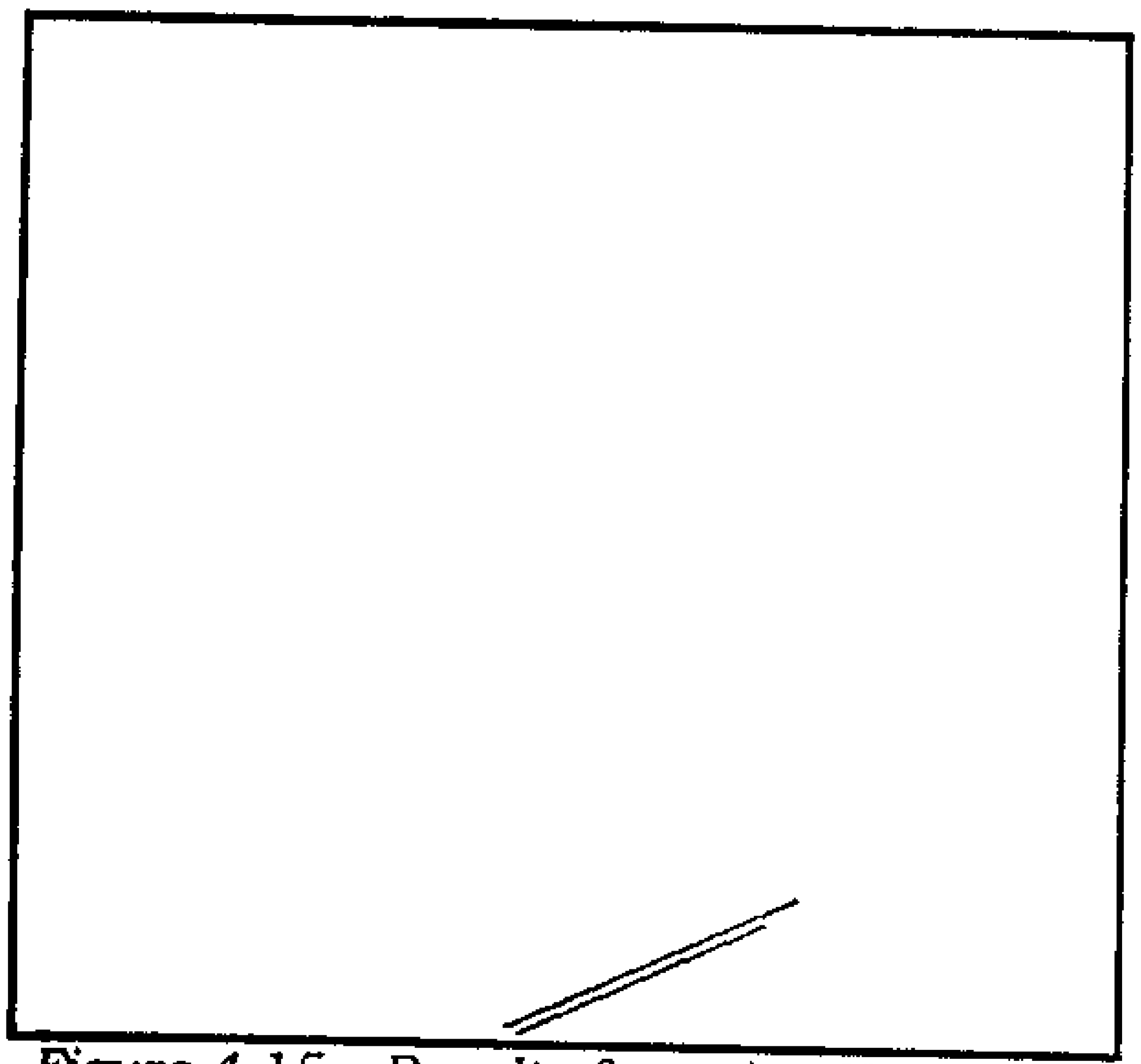


Figure 4.15 – Result of running Hough Transform on Figure 4.14 and reconstructing the lines.

Step 6 – Recognizing Villages/ Human Habitations : The image left after subtracting all the above recognized features consist essentially of villages, towns, or parts thereof, barring a few small sized components. (These components may remain because of noise, or because attempt has not been made to recognize all possible features in the red layer.) After these small-sized components are filtered out using area criterion, the minimum bounding octagon is determined for the remaining components and drawn on the E-map along with the symbol 'V'. The components recognized as villages are shown in Figure 4.16.

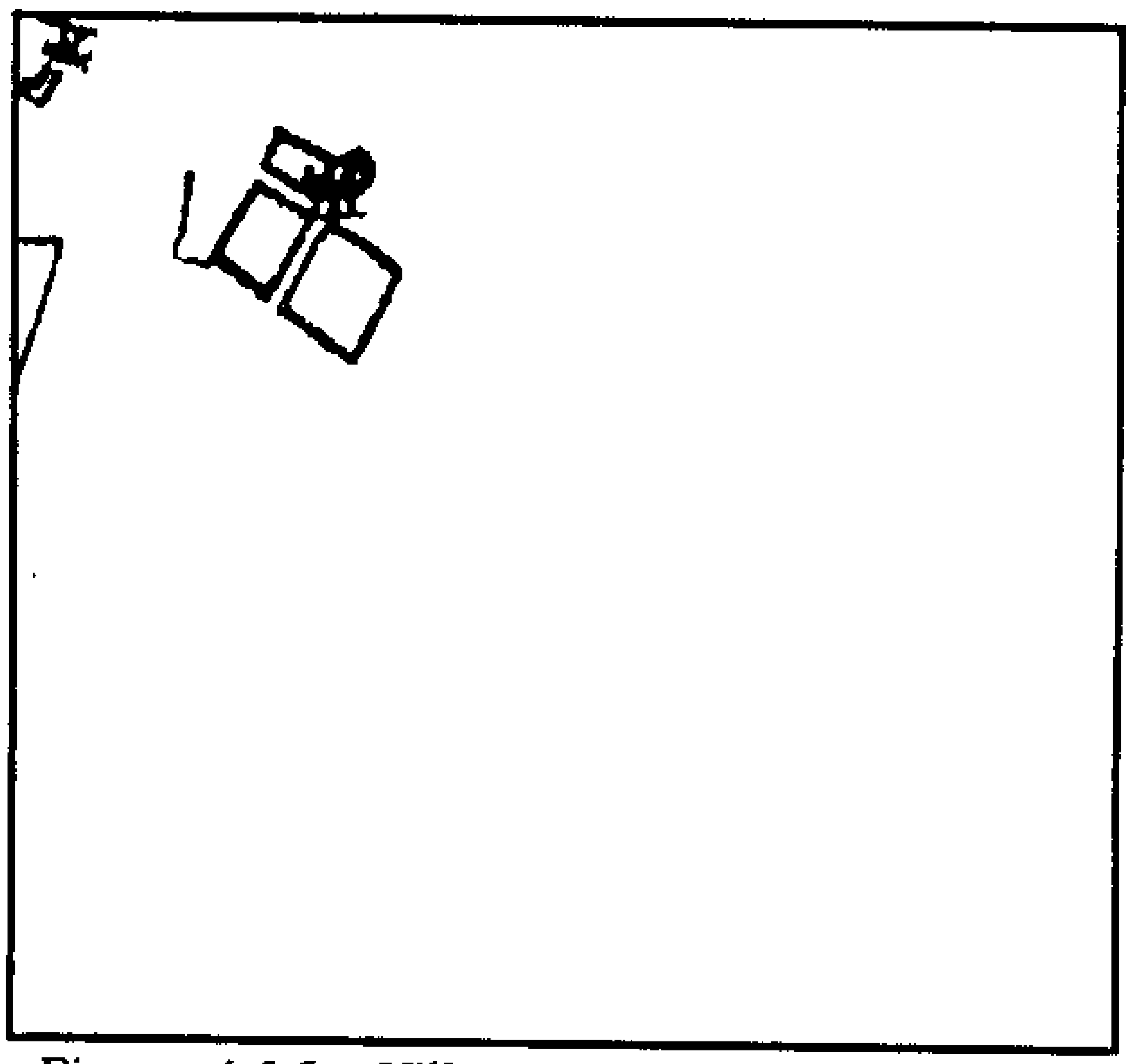


Figure 4.16 – Villages extracted from the training image

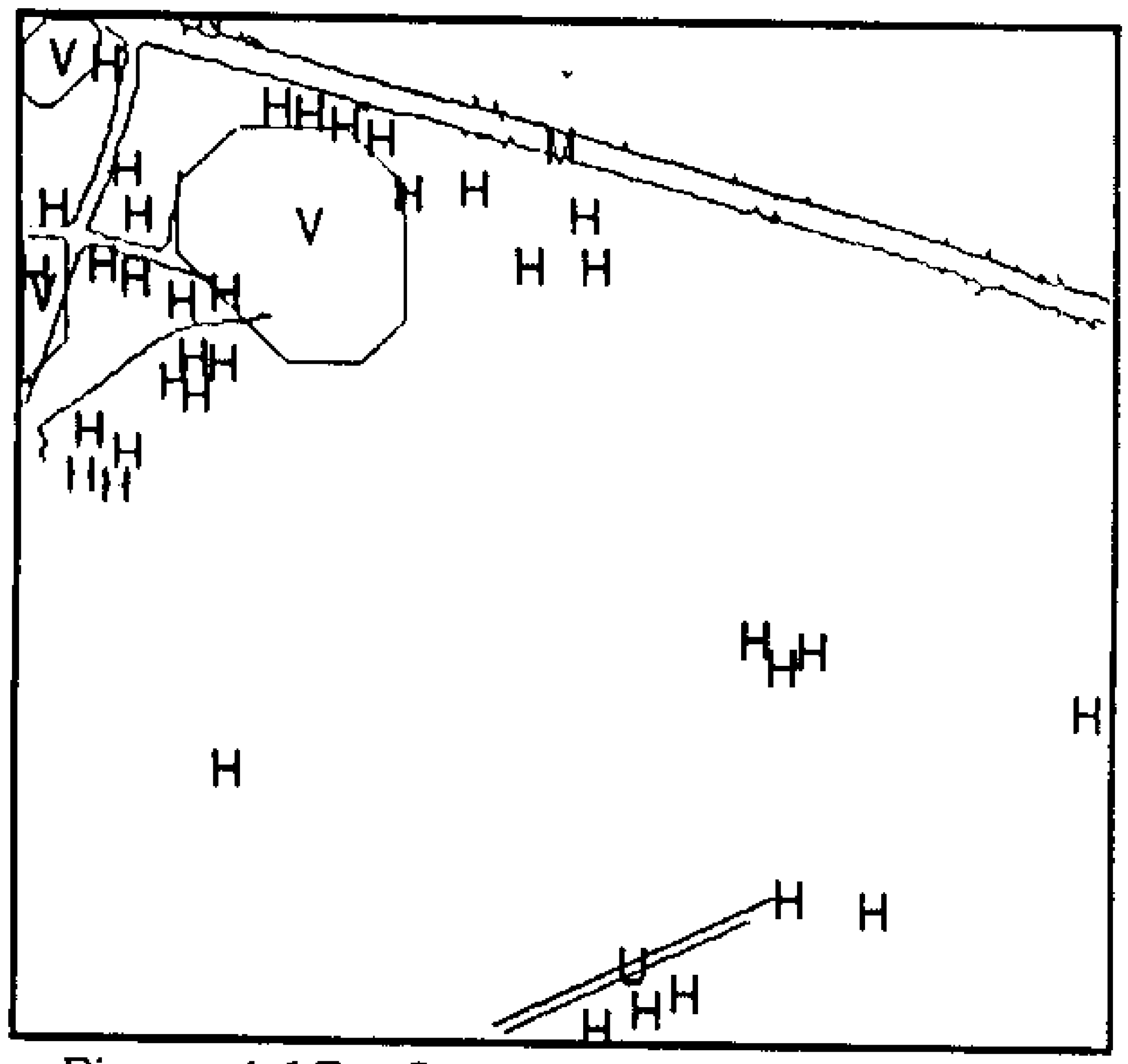


Figure 4.17 – Output E-map for the red layer of the training image.

The final output of the recognition and E-map generation phase on the red layer of the training image is shown in Figure 4.17.

4.6 Text Extraction from the Black Layer

The image of the black layer of the map consists of text strings, symbols representing telegraph land power lines, wells and tube-wells, dams, district and other boundaries, latitude and longitude lines, as well as outlines of water bodies and roads or railways. The aim here is to extract only the most important feature – text strings, so that the names can be recognized by an OCR system and entered into GIS.

To this end, size criterion similar to that mentioned in [18] is used here to judge components as character candidates. The specific steps are detailed below.

Step 1 – **Preprocessing :** As before, component labelling is done and components with very small area are removed. Although this removes positive noise, some amount of negative noise still remains in the form of holes in objects. This is removed by morphological closing operation.

The result of performing preprocessing on the black layer of the training image (Figure 3.3) is shown in Figure 4.20.

Step 2 – **Text Extraction using Size Criteria :** All characters have a fixed width and height. However, this may vary with font size and orientation. Because characters in maps can have any direction, both width and height are used.

In order to get a threshold to make a judgement, the size (the dimensions of both the sides the bounding box) of the components are counted and a histogram describing the size distribution is made.

A typical histogram expected from such a procedure when the map contains letters of only one font size is shown in Figure 4.18 [18]. There are three peaks in the histogram. The first peak 'a' is caused by numerous specks and by isolated lines in the image. The other peaks 'b' and 'c' represent the width and height respectively. From this distribution, a threshold may be easily selected for character candidates.

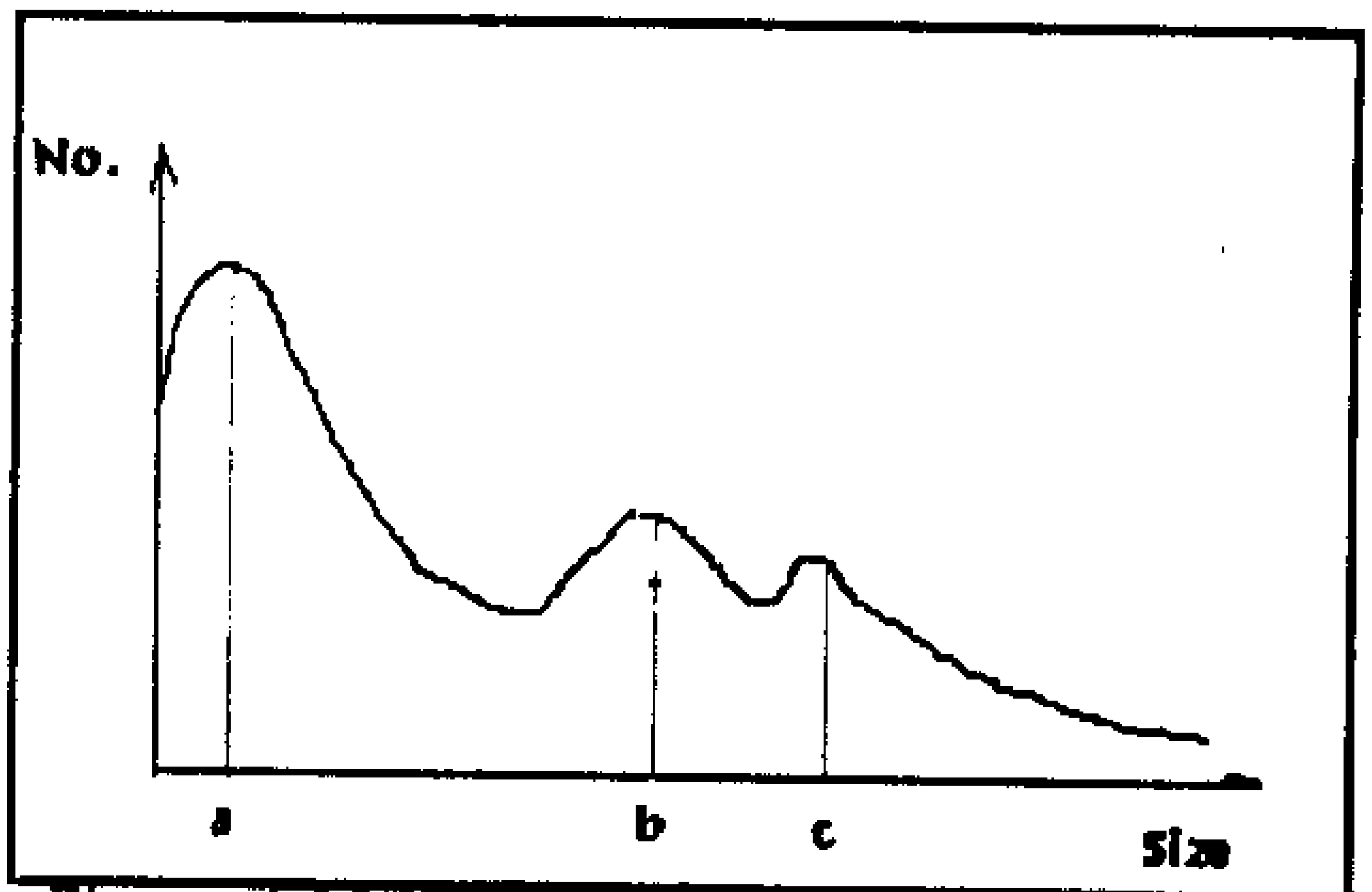


Figure 4.18 – Histogram for component size (ideal)

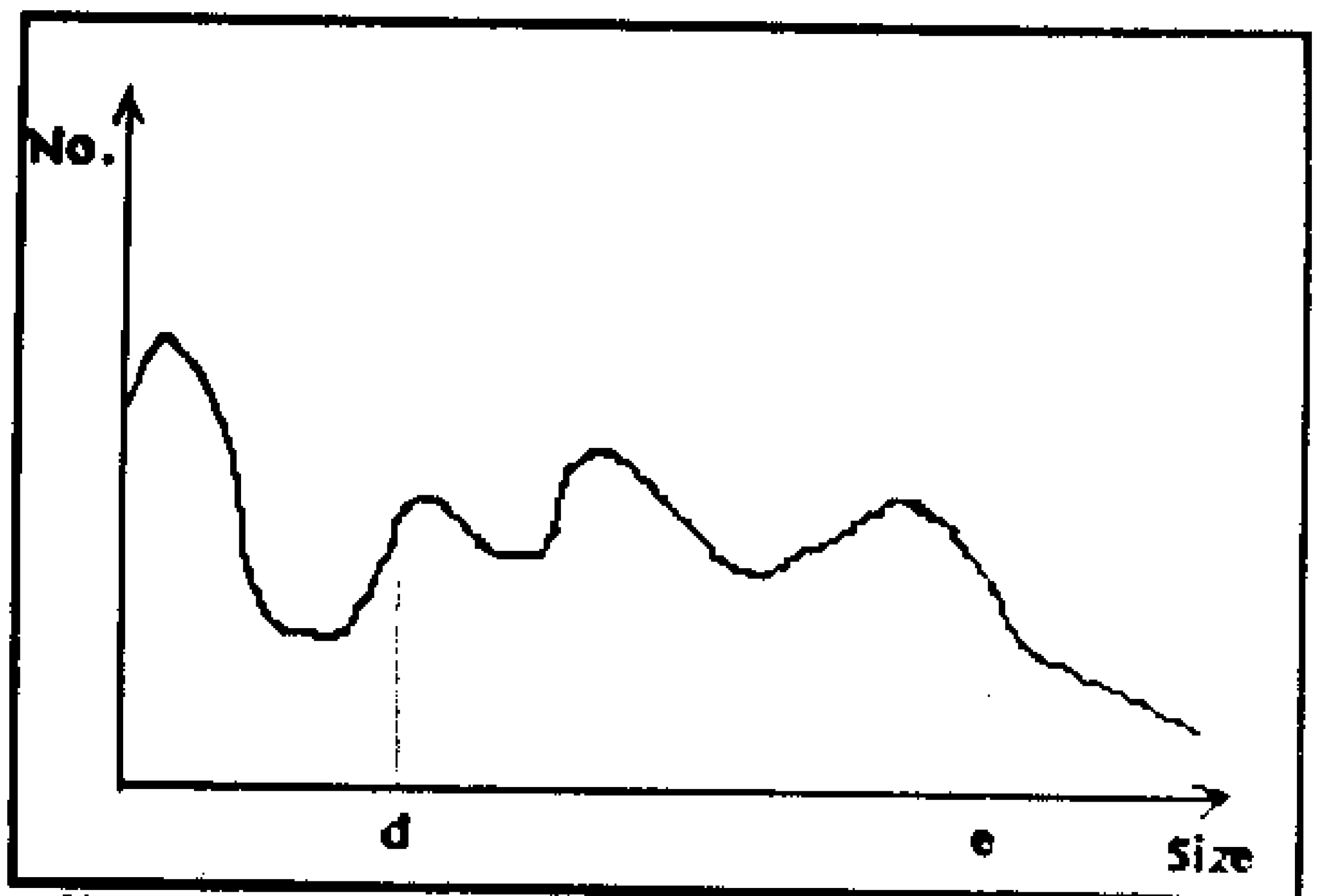


Figure 4.19 – Histogram for component size (actual)

However, since the present map contains fonts of different sizes, the histogram is not exactly trimodal, but of the form as shown in Figure 4.19. From this figure, characters may still be extracted by selecting components that lie within a range of peaks. These peaks correspond to the height or width of different font sizes. Thus all components that have at least one of the dimension of their bounding boxes between the sizes 'd' and 'e', are selected as characters.

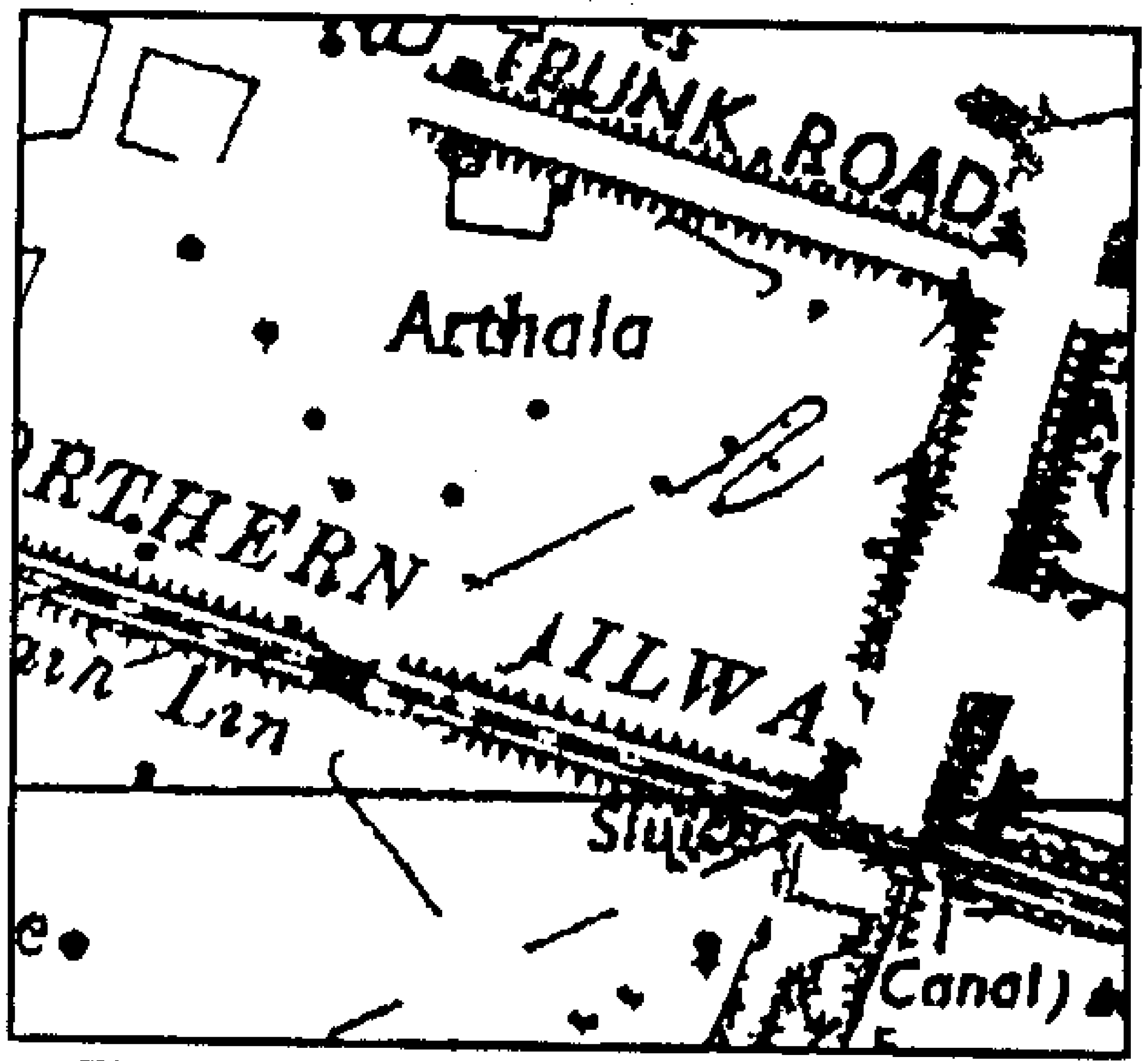


Figure 4.20 – Image of black layer of training image after preprocessing.

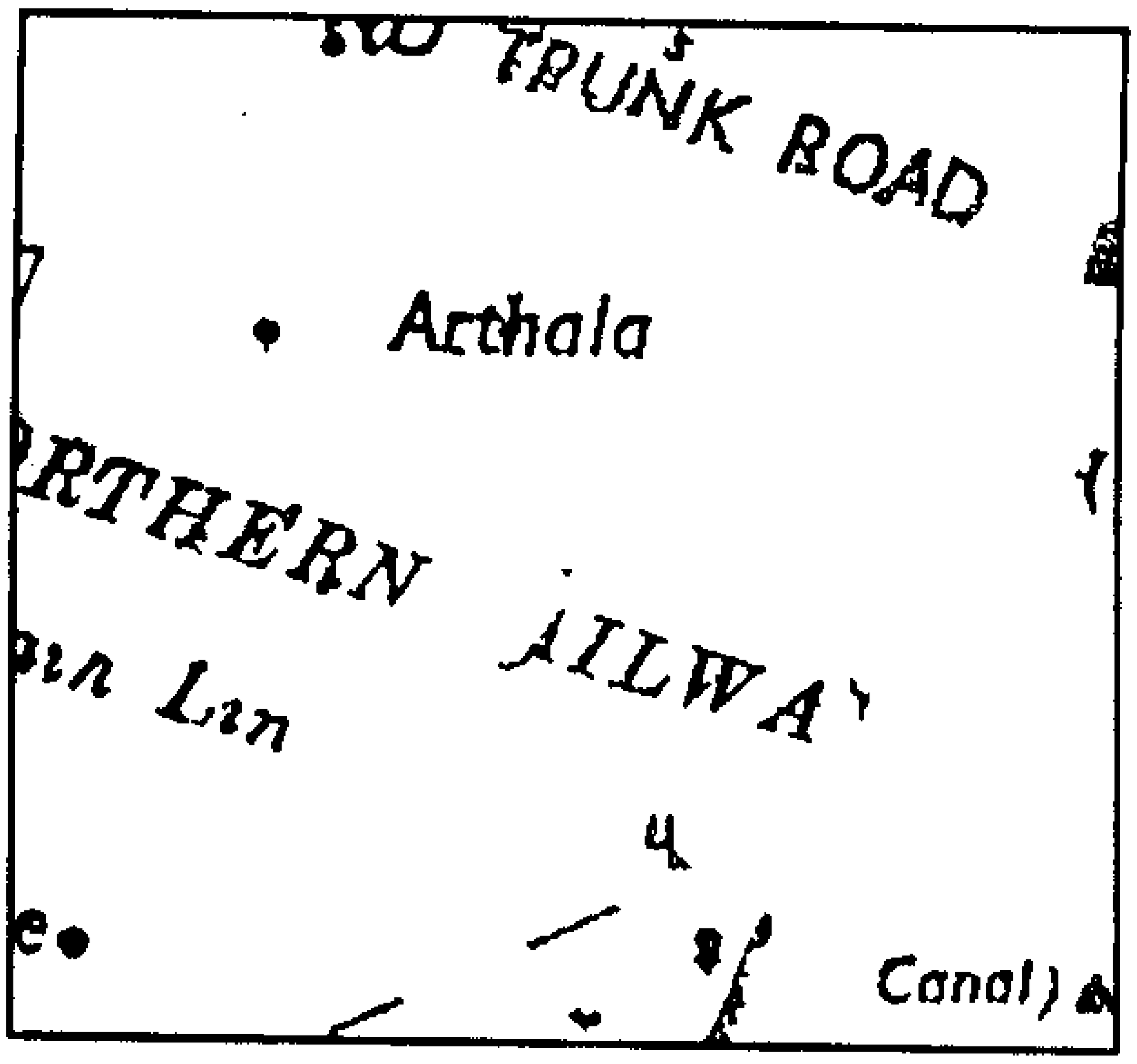


Figure 4.21 – Image after extraction of text from Figure 4.20.

The result of text extraction from Figure 4.20 is shown in Figure 4.21. Although a few non-text characters have crept in, the letters have almost been successfully isolated.

****

Chapter 5

RESULTS

5.1 Results from the Training Image

After describing in details the methodology followed in feature extraction from maps, it is now time to inspect the results and draw some conclusions about the performance of the said procedure. Let us first start with the training image.

The training image and all other test images were taken from the Survey Map 53H/6 showing parts of Delhi, Ghaziabad (U.P.) and Faridabad (Haryana) prepared by the Survey of India. The map was scanned at 300 dpi, giving a 2024 × 2035 pixels image. From this map, the training image (Fig 3.1) of size 400 × 380 pixels was chosen.

The images of the training image as it went through the various stages of this work has been shown along with the concerned description (separated layers in Chapter 3 and steps of recognition in Chapter 4). The final E-Map for this image (after combining Figures 4.7, 4.9, 4.17 and 4.21) is shown in Figure 5.1 (actual size). The facility used for this purpose was a Silicon Graphics machine with IRIX 5.6 as the operating system.

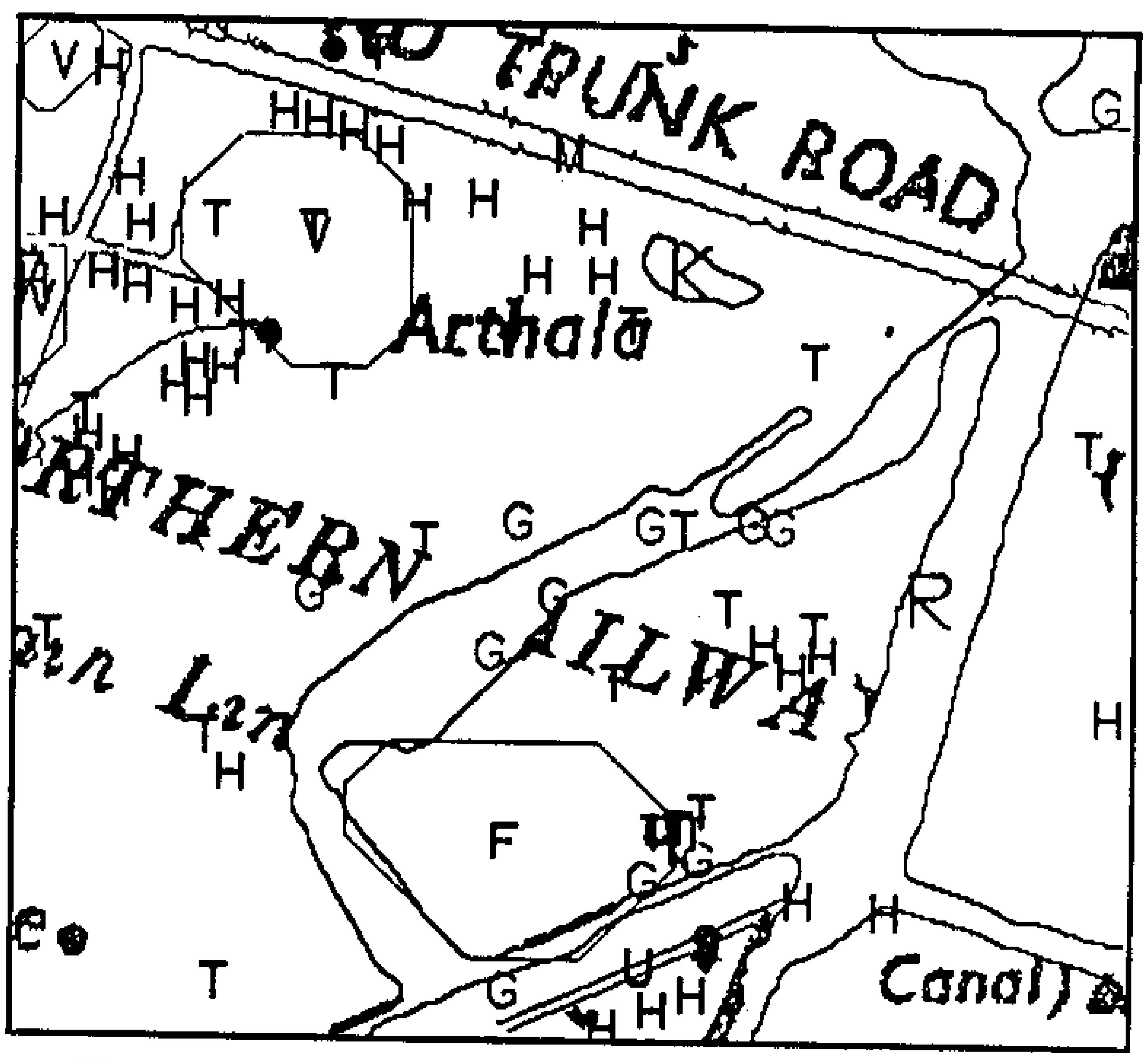


Figure 5.1 – Output E-Map for the Training Image.

Table 5.1 – Comparison of the number of features in the actual map and that in the E-Map

Feature	Number present in the 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. Metalled Road	1	1
8. Unmetalled Road	1	1
9. Cart Road	0	0
10. Village/Town	3	3

For the purpose of judging the accuracy of recognition of the features, a count of the different features as seen in the original training image and the output E-Map has been made. The result of this count, shown in Table 5.1, is a testimony of the high degree of accuracy of the map recognition process. In fact, the recognition rate is seen to be as high as 90%. The discrepancy in the number of trees and grasses resulted from the fact that some parts of the large green field stayed as separate components, and were thus recognized wrongly. The disparity in the number of huts is attributed to the presence of other features like dams, which were not treated as separate features.

Another measure of evaluation would be to use the following three error categories [13], common in optical character recognition (OCR). These are :-

(i) **Substitution errors** – in which a valid input symbol is assigned an incorrect valid classification (e.g., grass instead of tree);

(ii) **Deletion errors** – in which a valid input symbol is classified as undefined (e.g. X in place of a tree); and

(iii) **Insertion errors** – in which an invalid input symbol is assigned one of the valid classifications rather than being classified as undefined (e.g. hut instead of a symbol which actually represented part of a dam).

Using the above evaluation criteria, the following results are obtained.

Table 5.2 – Count of errors

Substitution errors	1	one tree was wrongly identified as grass
Deletion errors	1	one tree could not be recognized
Insertion errors	2	two components which are actually parts of a dam were recognized as huts

This count do not include the errors of counting four trees and three grasses for components that are actually part of the green field but recognized separately because they remained as separate components in the image.

5.2 Results for other Test Images

The map recognition procedure was performed on three other test images, all drawn from the same map as the training image. The results are shown in the following pages. (Figures 5.2 – 5.19). The count of errors for these images too is small, and is displayed in the following table. In all the three cases, recognition rates in excess of 90% have been recorded.

Table 5.3 – Count of errors for Test Maps I, II & III

	Test Image – I	Test Image – II	Test Image – III
Substitution errors	1	1	1
Deletion errors	0	4	4
Insertion errors	0	0	0

TEST MAP - I

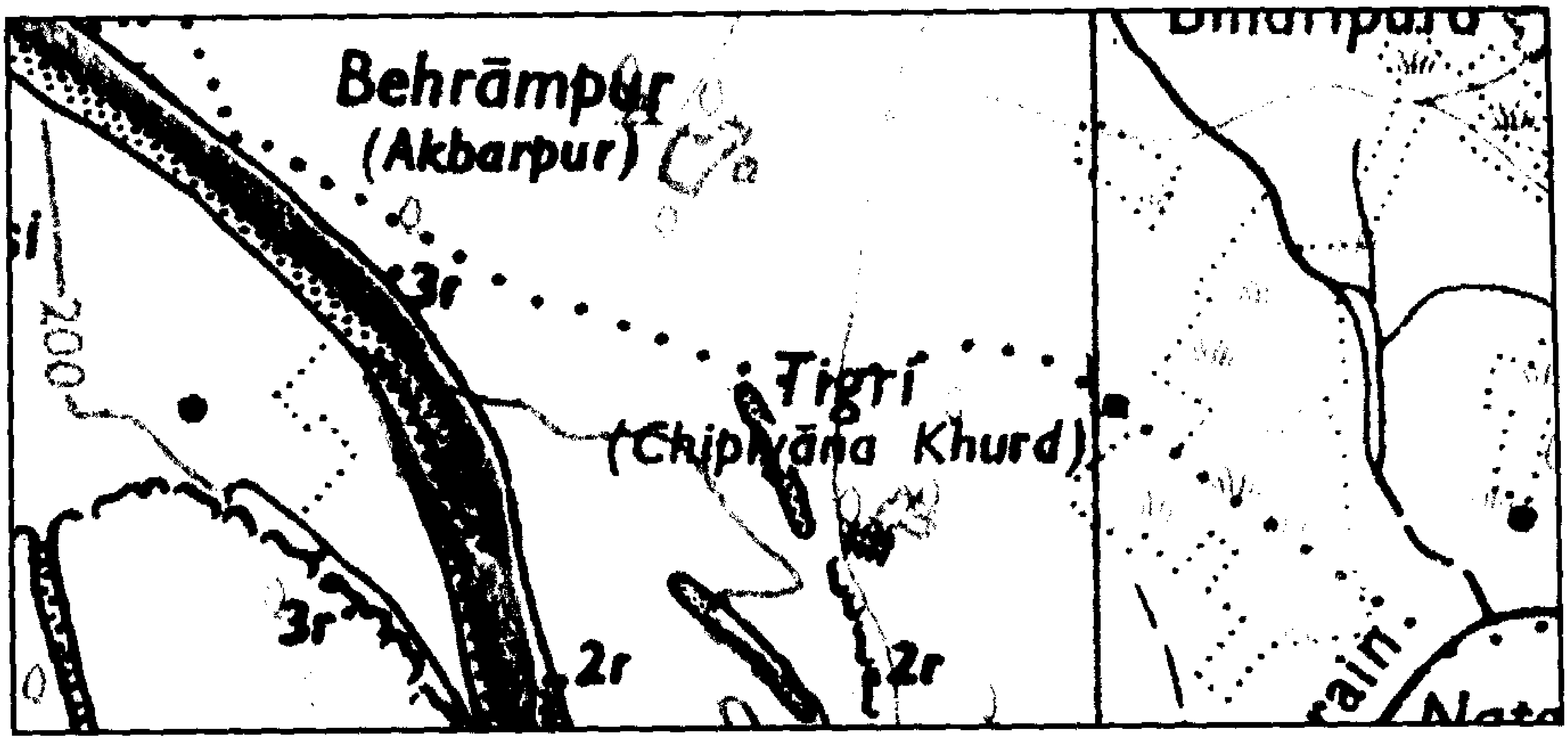


Figure 5.2 – Original Map

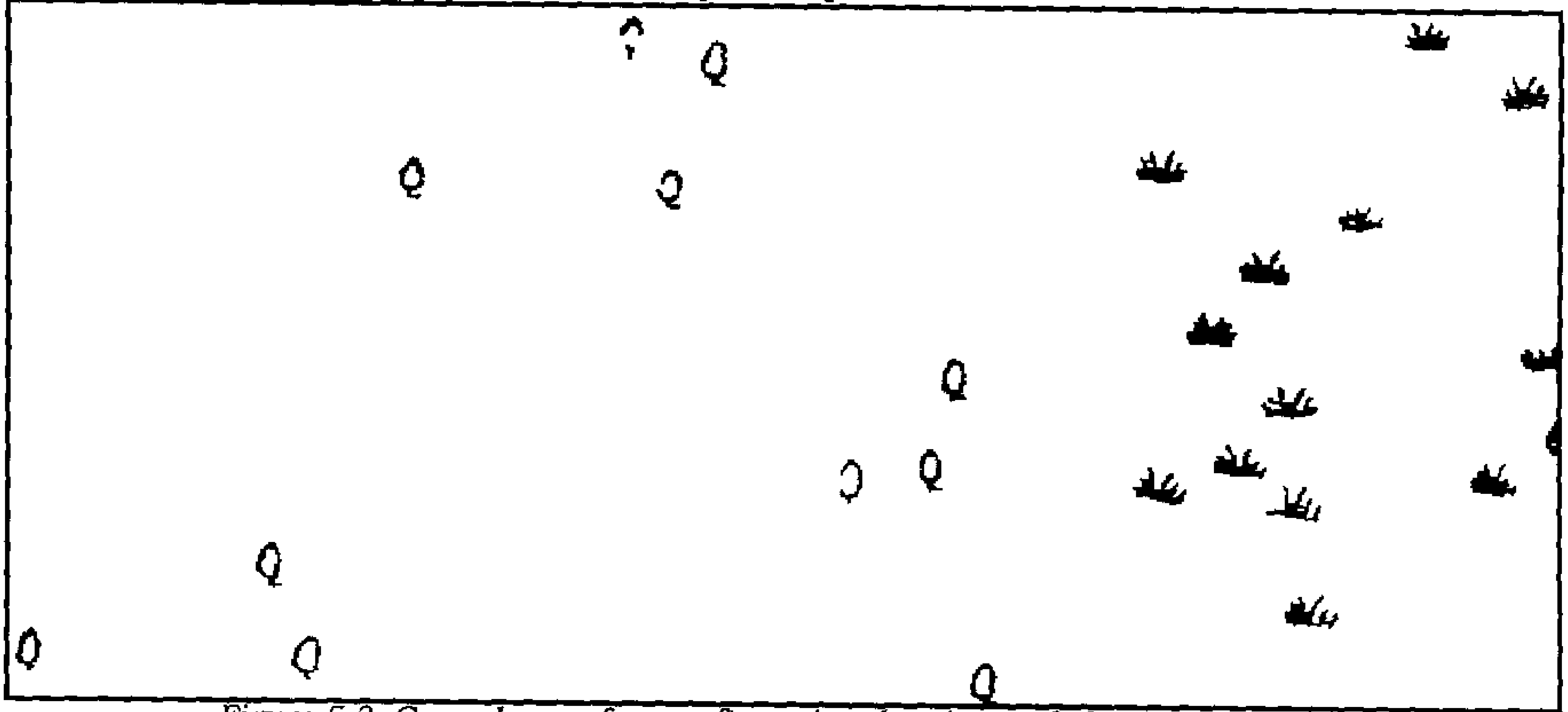


Figure 5.3- Green layer of map after noise cleaning and closing

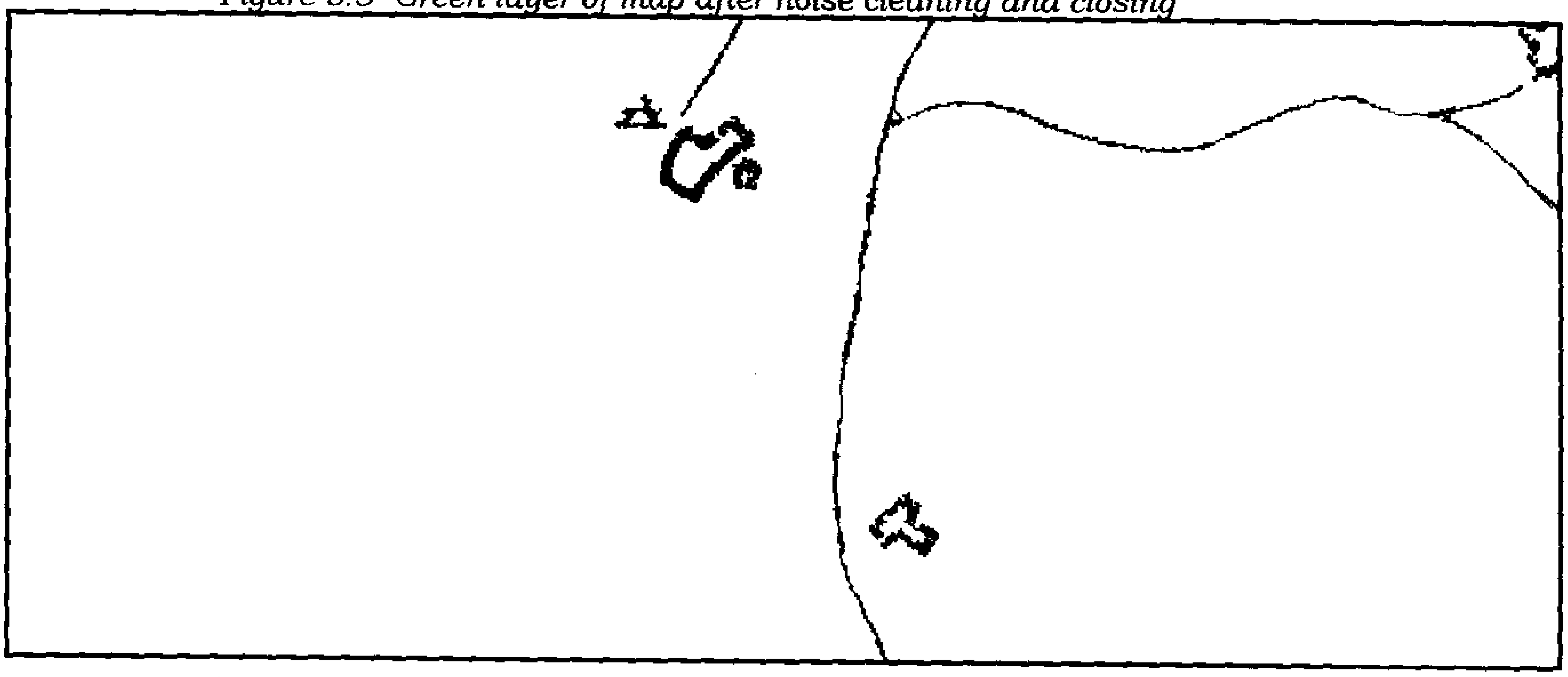


Figure 5.4 – Red Layer of map after noise cleaning and line joining

TEST MAP – I (contd.)

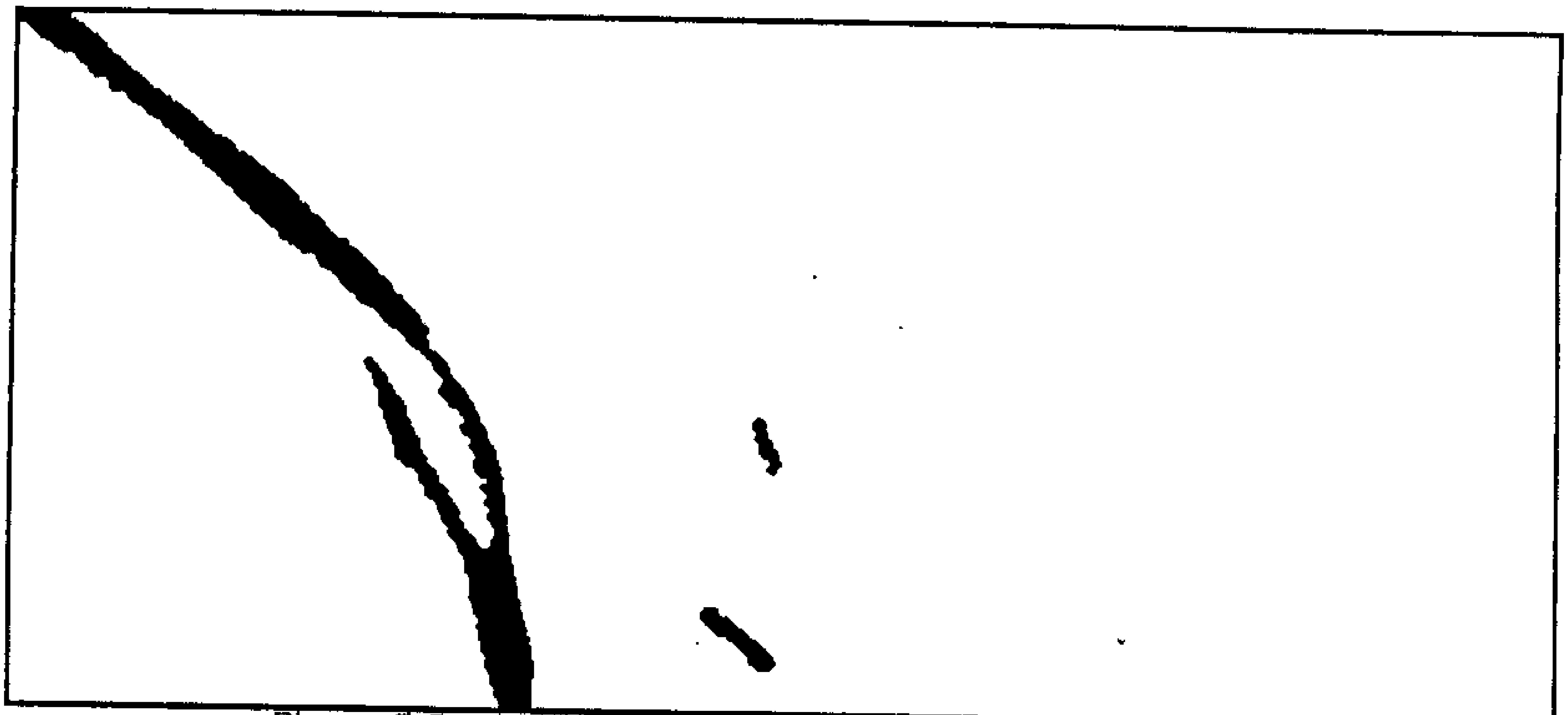


Figure 5.5 – Blue layer of map after preprocessing

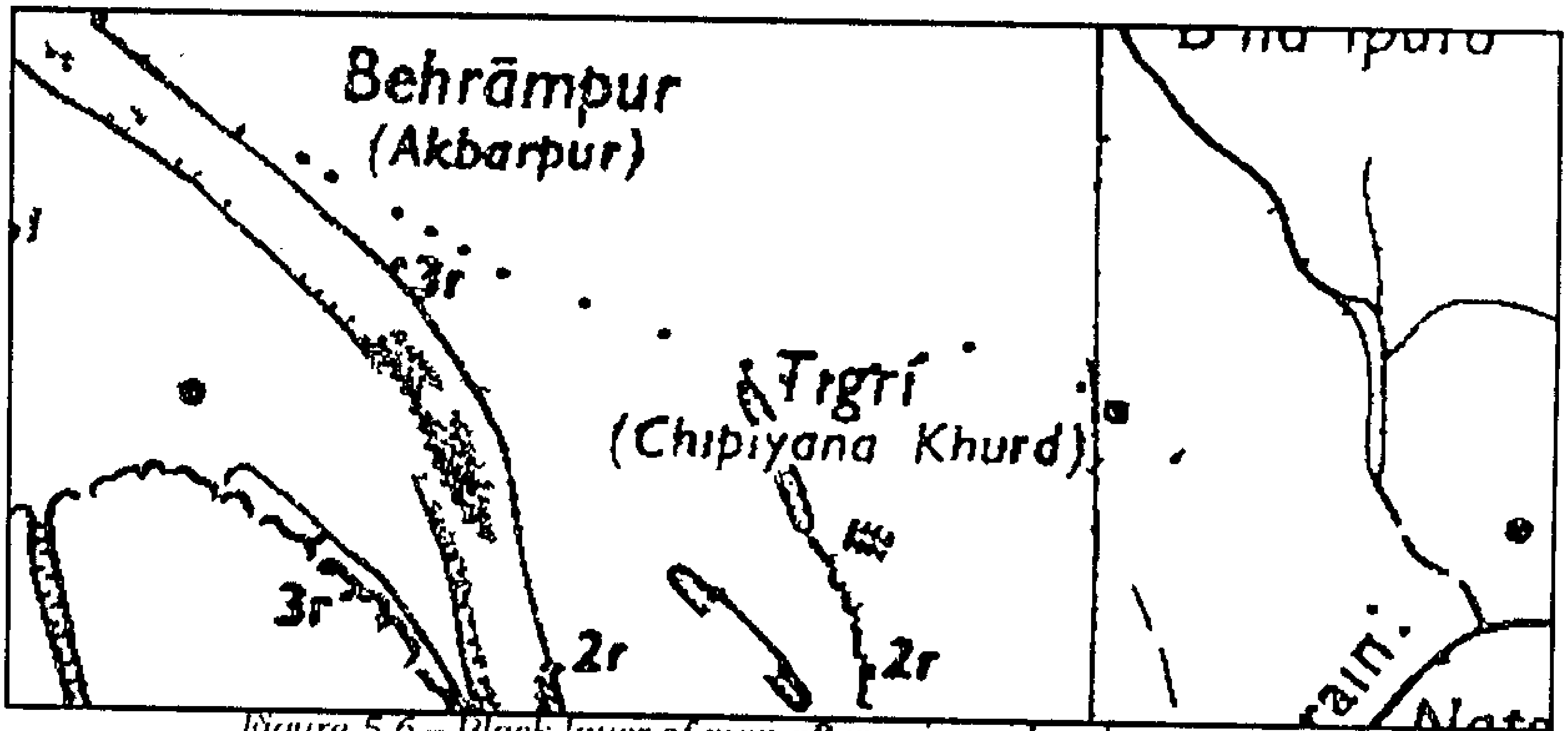


Figure 5.6 – Black layer of map after noise – cleaning

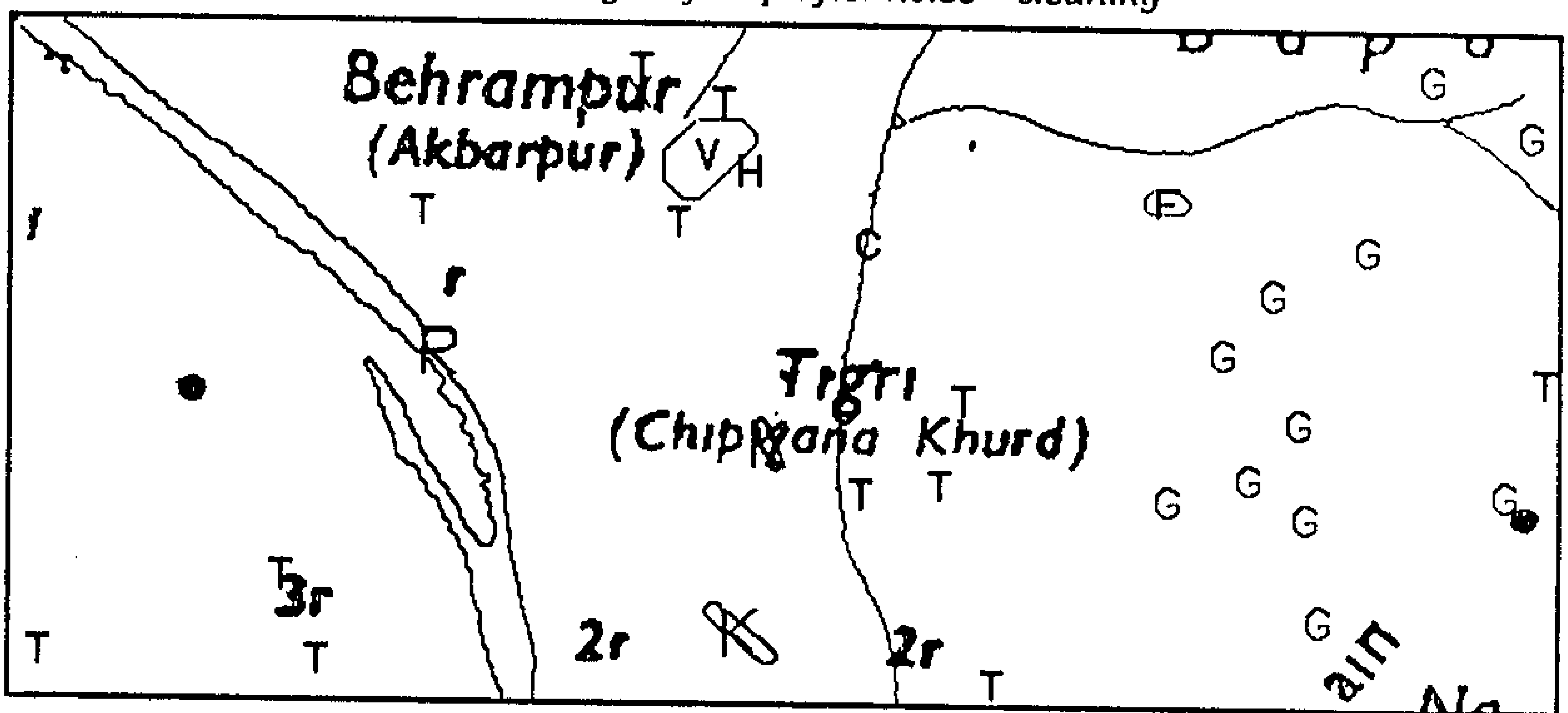
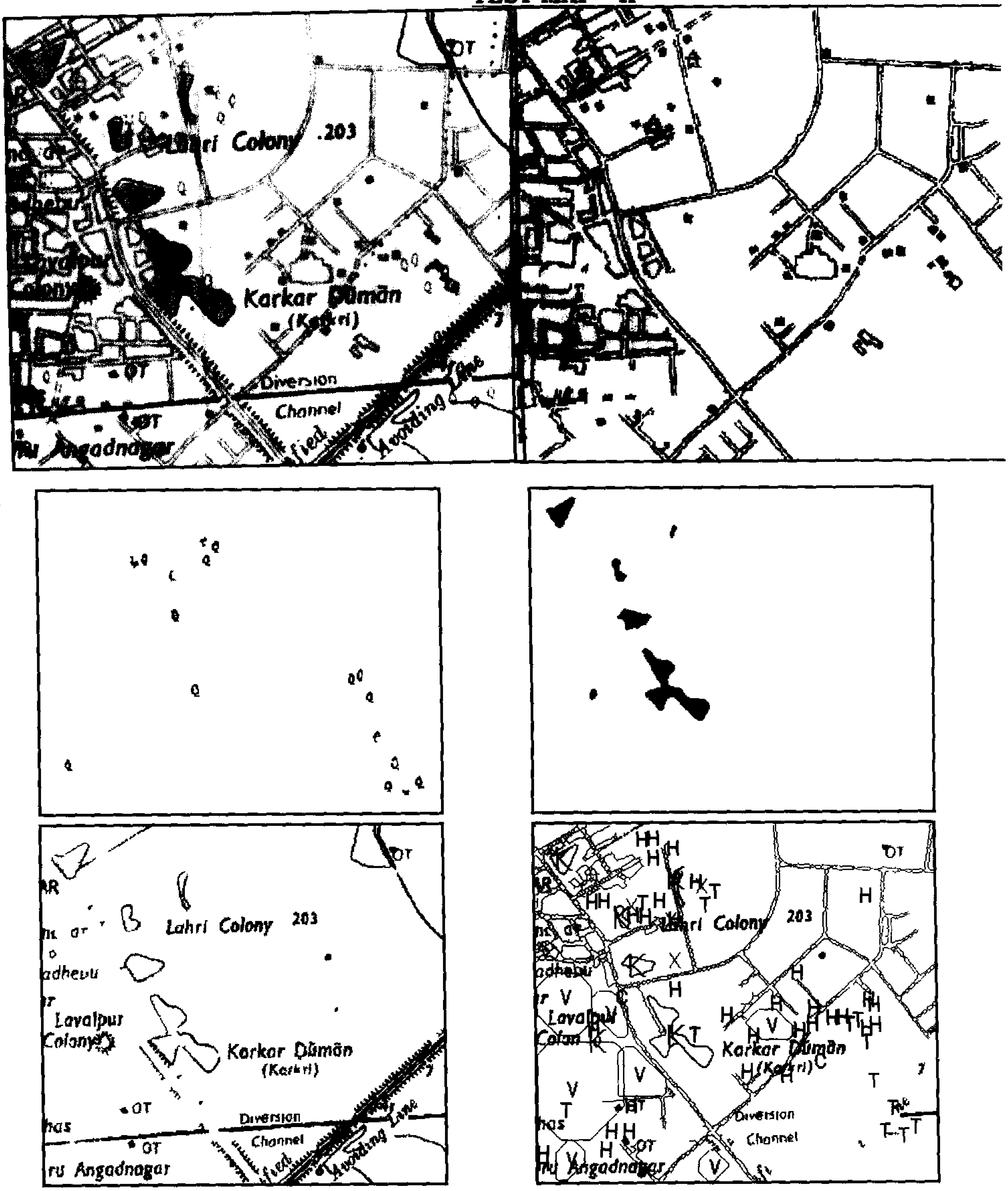


Figure 5.7 – Final E-Map generated for Test Map – I

TEST MAP – II



[1] *Figures 5.8 – 5.13 : Test Image II, its separated layers and output E-Map*

TEST IMAGE - III

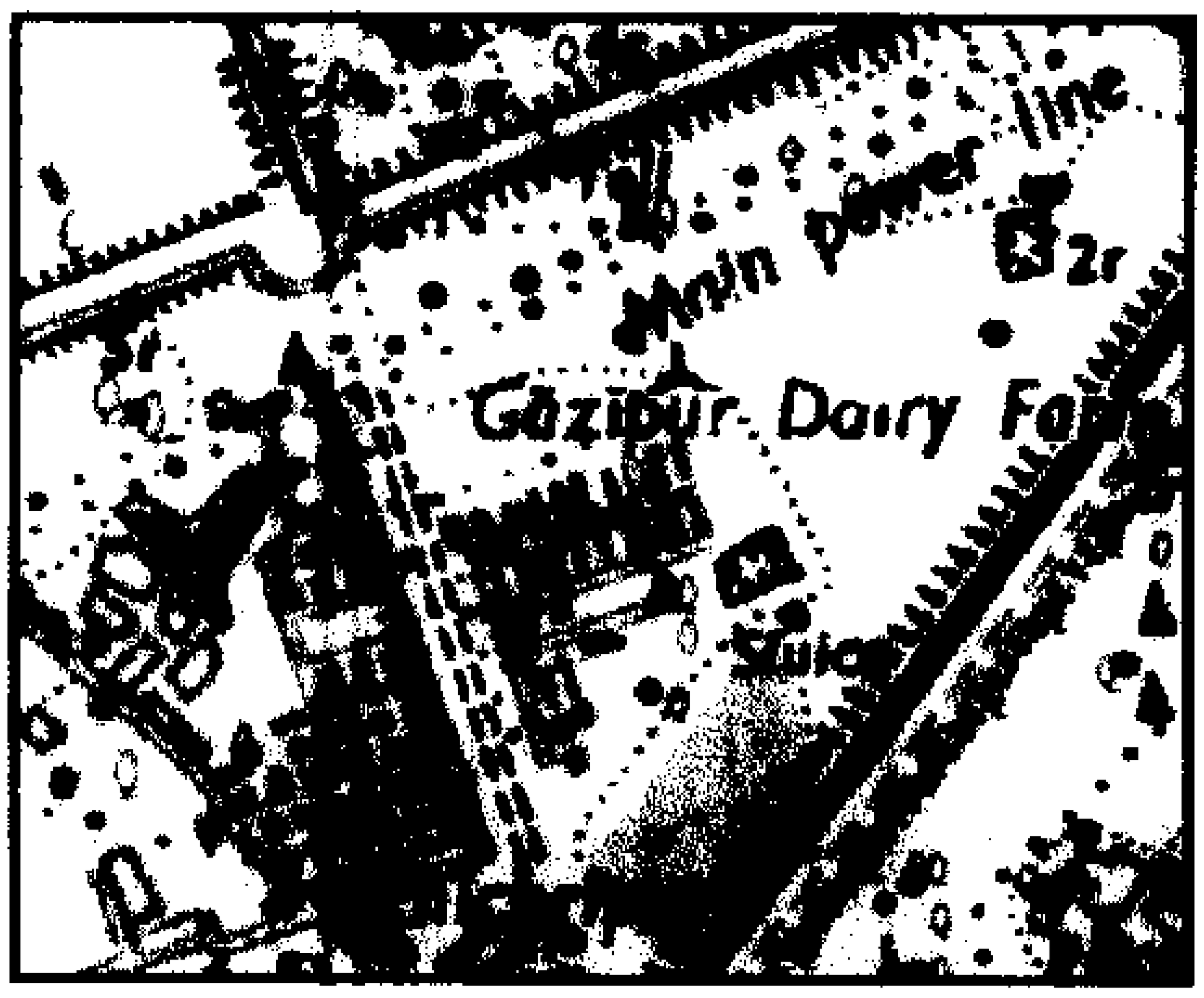


Figure 5.14 – Original Map

Figure 5.15 – Red layer

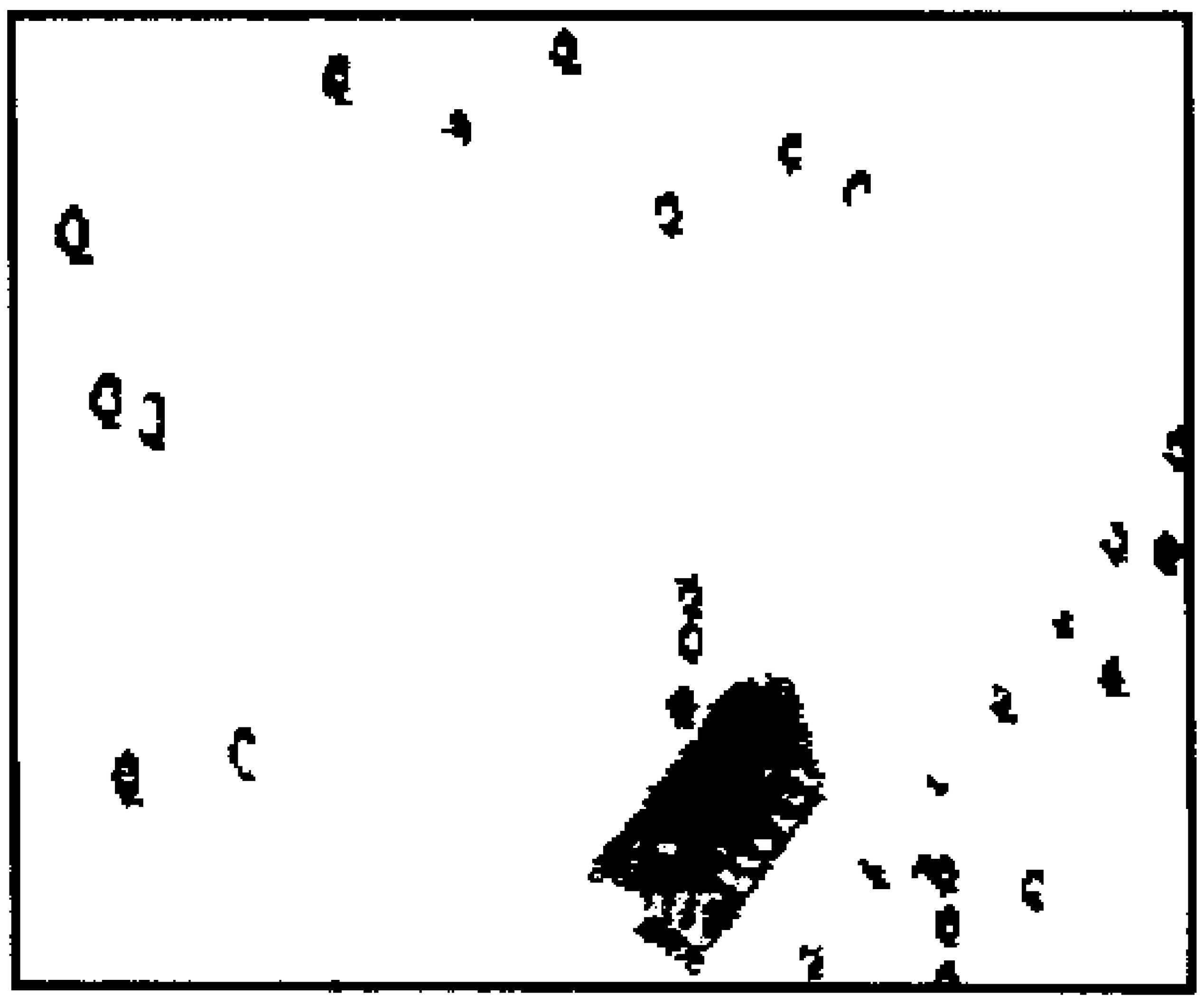


Figure 5.16 – Green layer

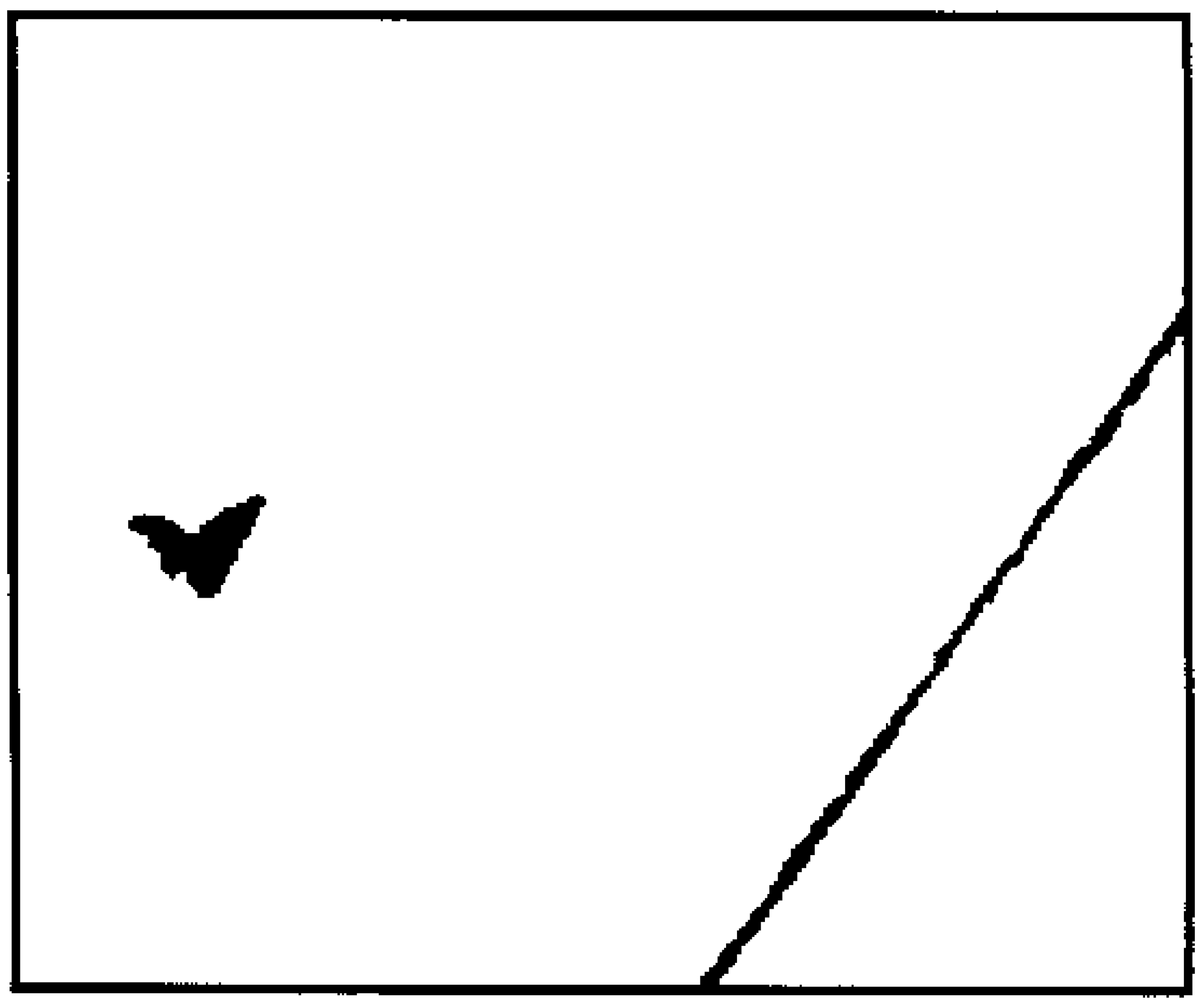


Figure 5.17 – Blue layer

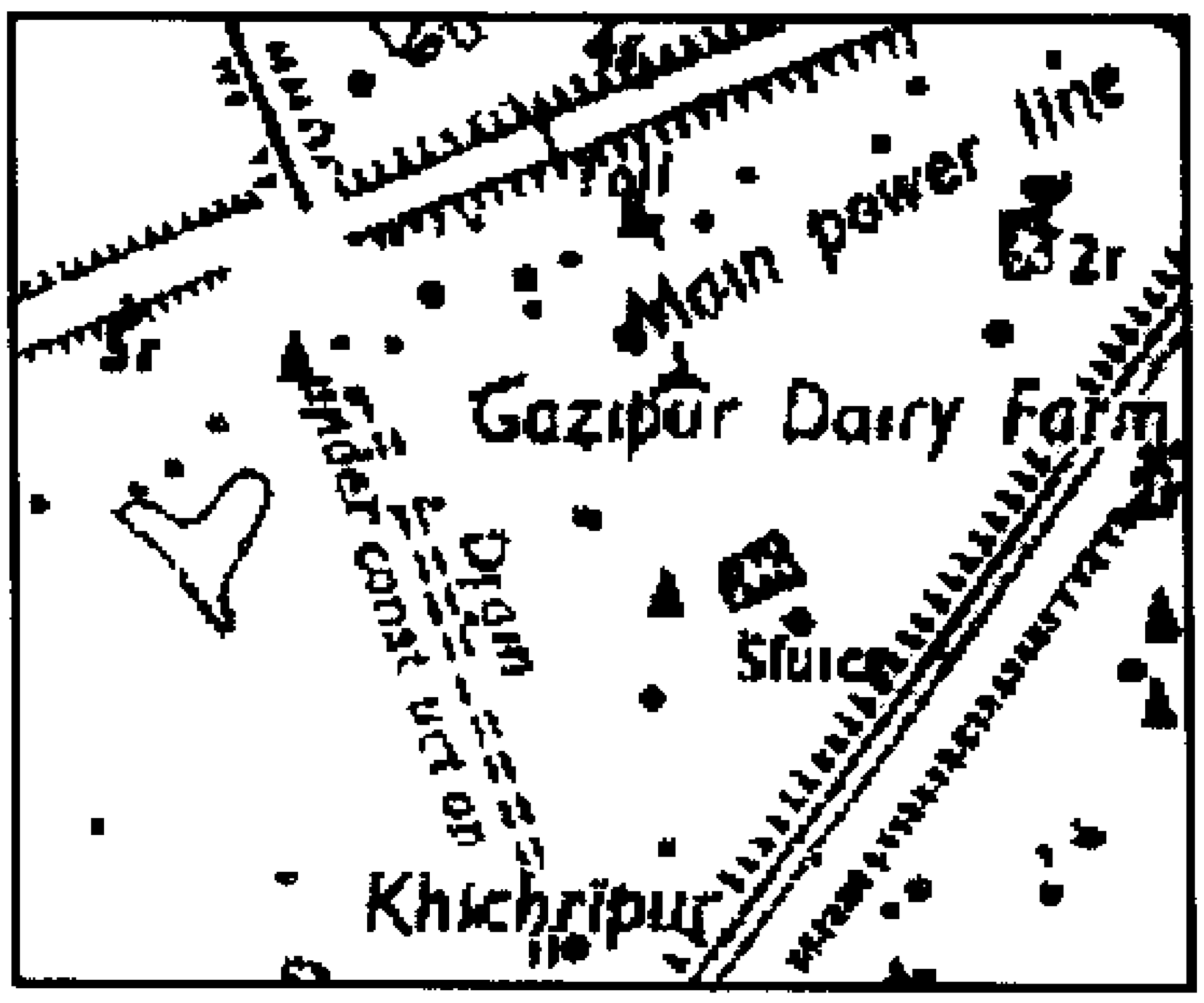


Figure 5.18 – Black layer

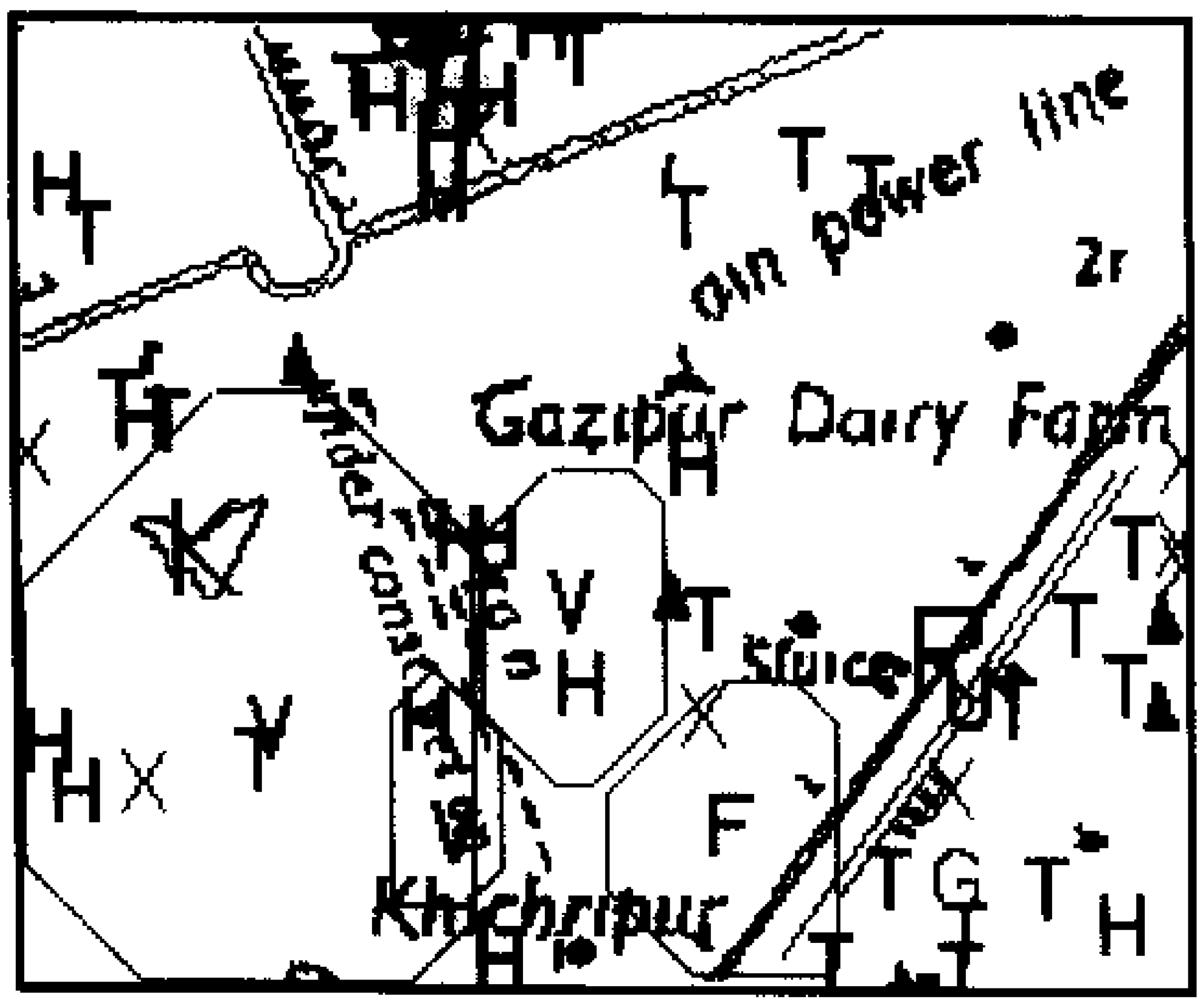


Figure 5.19 – Output E-Map

****

Chapter 6

CONCLUSION

6.1 Looking Back

As we come to the end of this report, it seems appropriate to cast a parting glance at the entire dissertation work. We have worked slowly and painstakingly at making topographic maps legible and understandable to the computer. Although it is simple for the human brain to interpret a map as the eye is viewing it, attempting to have the same work performed by the computer is exceedingly difficult.

The work has first focussed on separating the various layers a map is composed of. This was done taking advantage of the flat nature of the colours of the maps and using a clustering – classification approach. An extra enhancement operation was required for the initial training image. A pertinent observation in this direction is that these separate layers may not always correspond exactly to the original thematic layers.

The most challenging part of the work has undoubtedly been the design of a simple yet robust recognition procedure. None of the approaches adopted in literature – including template matching, legend-driven, statistical pattern matching principles seemed appropriate here. Instead, the focus remained throughout on 'discovering' simple spatial and geometrical attributes of the features to identify them even in the presence of noise. Naturally, recognition of all features found in maps was also not attempted. Consequently, the end of the work witnessed the extraction and recognition of features like tree, grass and field in the green layer; river and tank in the blue layer; hut, metalled road, unmetalled road, cart road, village, town and human habitations in the red layer; and separation of text strings from the black layer. As the precise locations of all these features, as well as the boundaries for fields, rivers, tanks and villages have been determined, it is well suited for use by GIS. Also, a visually appealing E-Map has been prepared for the purpose of viewing and comparison with the actual map.

Although quite a substantial amount of work has been done, we have no hesitation to admit that this has been only an introductory work in this field. A lot of work can still be done to enhance this system and its capabilities. Some thoughts in this regard are presented in the following section.

6.2 Scope for Improvement and Further Work

1. The most major impediment with the current implementation remains the amount of human intervention that is required. One way to reduce the quantity of manual supervision is to divide the operation of the system into a *'Learning Phase'* or *'User Verification Mode'* and an *'Automatic Mode'* [13]. In the learning phase, the user will input various parameters like threshold values, structuring elements of morphological operations, lines that need to be joined, and the like. All these user-input parameters will be stored, resulting in the build-up of a knowledge database to be used intelligently in the automatic phase.

2. Here the criteria for distinguishing between various features are developed based on extensive observation and intuition. Instead, it would be more useful if a systematic method based on Pattern Recognition or Artificial Intelligence principles could be constructed for symbol recognition. In the worst case, global descriptors like first and second moments, circularity, eccentricity and aspect ratio, as well as local shape descriptors like vacant ratio, Connectivity numbers and horizontal & vertical gaps per total area could be used as components of a feature vector.

3. The number of features recognized should be increased. The new features that can be added include contour lines, latitude and longitude lines, national, state and district boundaries, bridges, dams, wells and tube-wells, tramways, telegraph and power lines, temples, churches and mosques, hospitals and the like. Quite a number of the features consist of representations in two or more layers, for example submerged rocks are shown in black and blue. Interpretation of such features require particular care. Some features may be present in different layers depending on varying circumstances, for example, a dam may be represented in red or black depending on whether it is permanent or earth-filled. Such features should be tackled with a different strategy.

4. If the number of features that the system is capable of recognizing can be increased, the need for arranging them in an efficient data structure will arise. The features can be used to build up a training set library for subsequent efficient retrieval and

recognition. Samet and Soffer [12, 13, 14] have used *adaptive k-d trees* for the purpose.

5. The difficulty due to obstruction of features in different layers is a big problem. After layer separation, several of the features get 'cut out' resulting in difficult-to-recognize features. A method must be devised to have this layer separation process without loss of information or to somehow obtain separate layers of maps. For example, a method is described in [1] to generate images of isolated objects when the image consists of the silhouettes of two objects overlapping each other. Efforts could be made to enhance the method to solve the present problem. If this could be done, user intervention, for example, joining of rivers under manual supervision, could be reduced significantly.

6. For recognition of a complete map, to reduce computational time, the map may be divided into 'tiles' of manageable sizes, say 512 × 512 pixels, each tile processed separately, and the results combined. This method would be advantageous if several processors are available so that parallel processing could be done.

7. A final thought involves the integration of Query-processing with the present system in order to enhance its functionality. For example, it would be wonderful to have the computer answer a query like 'List all hospitals that are within a 3 km radius of such-and-such hotel.' Vigorous research is being pursued in this area, both regarding the nature of query and its format, and also the procedure for processing the query. Though natural language is the obvious choice for users to make the query, work in this field has been limited to SQL-type or LISP-based queries having a predefined syntax.

BIBLIOGRAPHY

Books

1. Chanda, B. and Dutta Majumdar, D. – *Digital Image Processing and Analysis*, Prentice Hall of India, New Delhi, 2000.

2. Gonzalez, R.C. and Woods, R.E. – *Digital Image Processing*, 3rd ed., Addison-Wesley, Reading, Massachusetts, 1999.

3. Rosenfeld, A. and Kak, A.C. – *Digital Picture Processing*, 2nd ed., Vol. 2, Academic Press, New York, 1982.

Journal Papers

4. Peuquet, D.J. – 'An examination of techniques for reformatting cartographic data, Part I – The raster-to-vector process.' *Cartographica*, 18(1): 34-48, January 1981.

5. Ablameyko, S.V., Beregov, B.S. and Kryuchkov, A.N. – 'Computer-aided cartographical system for map digitizing'. *Proceedings of the Second International Conference on Document Analysis and Recognition*, pp. 115-118, October 1993.

6. Starr, L.E. – 'Computer-assisted cartography research and development report', *International Cartographic Association*, July, 1984.

7. Ejiri, M., Kakumoto, S., Miyatake, T., Shimada, S. and Ichikawa T. – 'Automatic recognition of design drawings and maps', *Proceedings of the Seventh International Conference on Pattern Recognition*, pp. 1296 – 1305, 1984.

8. Malerba, D., Esposito, F., Lanza, A. and Lisi, F.A. – 'First Order Rule Inductionfor the Recognition of Morphological Patterns in Topographical Maps', In : Perner (ed.) *Proceedings of the Second International Workshop on Machine Learning and Data Mining in Pattern Recognition, Lecture Notes in Computer Science*, Vol. 2123, Springer, Germany, July 2001.

9. Kasturi, R. and Alemany, J. – 'Information extraction from images of paper-based maps', *IEEE Transactions on Software Engineering*, 14(5): 671-675, May 1988.

10. Dupon, F., Deseilligny, M.P. and Gondran, M. – 'Automatic Interpretation of Scanned Maps : Reconstruction of Contour Lines', In : Tombre and Chhabra (eds.) *Graphics Recognition : Algorithms and Systems, Lecture Notes in Computer Science, Proceedings of the Second International Conference*, Vol. 1389, Springer – Verlag, Berlin, 1998.

11. Yamada, H., Yamamoto, K. and Hosokawa, K. – '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.

12. Samet, H. and Soffer, A. – 'A legend-driven geographic symbol recognition system', *Proceedings of the Twelfth International Conference on Pattern Recognition*, Vol. II, pp. 350-355, October 1994.

13. Samet, H. and Soffer, A. – 'MAGELLAN : Map Acquisition of Geographical Labels by Legend Analysis', *International Journal of Document Analysis and Recognition,* 1(2) : 89-101, June 1998.

14. Samet, H. and Soffer, A. – 'MARCO: MAp Retrieval by COntent', *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 18(8) : 783-798, August 1996.

15. Lauterbach, B. and Anheier, W. – 'Segmentation of scanned maps into uniform colour spaces', *IAPR Workshop on Machine Vision Applications,* pp 322-325, 1994.

16. Ebi, N., Lauterbach, B. and Anheier, W. – 'An image analysis system for automatic data acquisition from coloured scanned maps', *Machine Vision Applications,* 7(3) : 148 – 164, 1994.

17. Fletcher, L.A. and Kasturi, R. – '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.

18. Gao, J., Tang, L., Liu, W. and Tang, Z. – 'Segmentation and recognition of dimension texts in engineering drawings', *Proceedings of the Third International Conference on Document Analysis and Recognition,* Vol. I, pp. 528-531, August 1995.

19. Luo, H., Agam, G. and Dinstein, I. – 'Directional Mathematical Morphology for line thinning and extraction of character strings from maps and line drawings', *Proceedings of the Third International Conference on Document Analysis and Recognition,* Vol. I, pp. 257-260, August 1995.

20. Deseilligny, M.P., Men, H.L. and Stamon, G. –'Characters String Recognition on Maps, A method for high level reconstruction', *Proceedings of the Third International Conference on Document Analysis and Recognition,* Vol. I, pp. 249-252, August 1995.

Internet Resources

21. OpenGIS Consortium : *The OpenGIS Abstract Specification(1996),* www.opengis.org/techno/abstract.html

22. Man, V.D. and Gerland, P. – 'Software Development and Recent Technological Innovations for Geographical and Statistical Applications for Population Activities – The UN Statistics Division Experience', *First International Health Geographics Conference,* October 1998, http://sauvy.ined.fr/seminaires/demodynamiques/98-99/gerland-text/jhupaper.html

23. Chao, H.M. and Trinder, J.C. – 'Feature extraction in residential areas by knowledge modelling', www.gisdevelopment.net/aars/acrs/2000/ts9/imgp0016pf.htm

24. Araphant, Pakorn – 'Automated cartographic line tracking', www.gisdevelopment.net/aars/acrs/2000/ps1/ps120pf.htm

25. The Practical Transformer – Guide to almost all known Image Processing Transforms including Hough Transform and Fourier Transform, http://perso.wanadoo.fr/polyvalens/clemens/transforms/index.html

●●●●●●●●●●●●●●●