Adaptive Altitude Feature Recognition for Paper based Topographic map

Sandhya Banda, Arun Agarwal, C.Raghavendra Rao, Rajeev Wankar
University Of Hyderabad
Hyderabad, India

***Abstract*—Contour layer of topographic maps provides useful information for many vector based GIS applications. Binary image of contour layer is obtained from a topomap by employing color clustering algorithms. The resultant contour layer contains altitude values in addition to the contours, as they are represented using the same color as contours. The method proposed helps to produce a clean contour layer and a layer of elevation values, which are connected to the contour lines. Filtering is done in two stages. EM clustering has been used to separate the text not connected to contours in the first stage. The results from first stage are used in the second stage to detect text connected to contours. The extracted text has to be recognized and tagged to the contour line, to generate a 3D model. Various challenges involved in recognition of altitude tags are discussed. The digits from the tags are segmented, preprocessed and recognized using a convolutional neural network, LeNet - 5.**

***Index Terms*—Topographic map; Contour layer; Filtering; Altitude tags; Connected components; EM clustering;Digit Recognition**

I. INTRODUCTION

Topomaps consist of colour line drawings that use linear and area features to portray information content [1]. A topomap is cluttered with entities which have multidimensional features. Various GIS applications demand tools for segregation, digitization and integration of topomap. Topomaps describe spatial data using entities, attributes and relationships in an unstructured representation. Entities on maps are described by point, line, and area objects. Symbols, Text, Color are commonly used on maps to represent attributes associated with the entities. Relationships among entities are depicted visually. A sample map is shown in 1. In this map, water bodies are shown in blue, toponyms in black, contour lines in brown. Our focus is on the generation of layer pertaining to contours, which has a tremendous application to geo scientific modelling due to the potential 3D representation of the terrain (which is a demanding tool from Geoscientists to take the advantage of visualization tools). Contour lines are represented by lines with 2-4 pixels width when scanned at a resolution of 300dpi, brown in colour. The lines are labelled indicating the elevation values. In [2] is presented various challenges of automatic extraction and vectorization of contour lines from colour topomaps. A layer corresponding to contours will invariably have noise, discontinuities and gaps. The extraction of a contour layer thus introduces a challenging problem of filtering unwanted text, identification of gaps and filling them, for realistic automatic vectorization. This paper proposes an approach to filter text and symbols from contour lines. The filtered altitude tags should be recognized to tag the contour lines with an altitude value and generate a 3D model. Hence the paper introduces recognition of altitude tags.

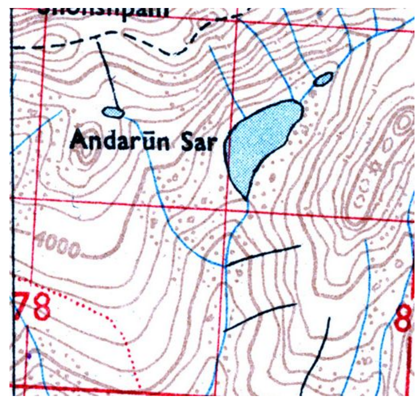


Fig. 1. Sample topographic map

The paper is organized as follows: A brief description of the related work in this area is given in Section II. Section III describes the various modules in processing of topographic maps. Section IV describes the proposed method for separating the contour layer and the altitude tags. Section V describes the steps in altitude tag recognition. Section VI shows the results on the contour map and gives Conclusion.

II. RELATED WORK

Research on generation of contour layer of a topomap focuses mainly in extracting contour lines using various color based segmentation methods, filtering and reconstructing them. When a contour layer is extracted from a color topomap, two problems occur. One is that the contour layer has noise due to improper segmentation in addition to altitude values and hence a Filtering process to clean the noise and segment the text from contours is essential. Second issue is that the contour lines have gaps and discontinuities due to intersecting and overlapping layers. A reconstruction module to identify and fill the gaps is necessary. This has been discussed in [3]. In this paper the filtering and recognition of altitude tags is presented. Section II-A gives a survey about the extraction of contour layer and Section II-B gives some work on separation of text from document images. The altitude tags belong to the class of digits and hence in Section II-C existing literature on digit recognition is reported.

A. Extraction of contour layer

Early segmentation methods on topomaps concentrated on color space selection. Ansoult et al. [4] use the mean and variance of the hue channel for discriminating soil types on a digitized soil map. Ebi et al. [5] transform the input RGB color space into another color space considering the chromaticity. Khotanzad et al. [2] have addressed the problem by proposing a method to build a color key set for linear and area features of standard USGS topomaps and a non parametric clustering algorithm based on multidimensional histogram of the topomap. Yang Chen et al. [6] have extended this work beyond the standard USGS maps to common conditioned maps by using the features extracted from the gray version of the same color map along with the color coded map. Carol Rids et al. [7] have used color edge detection - vector angle edge detector, with a saturation-based combination of hue and intensity planes, and color clustering (minimum variance quantization) to extract the contour lines from color coded maps.

B. Separation of text/graphics

Automatic segmentation of text/symbols from documents has been one of the fundamental aims in graphics recognition. One way has been to separate text from documents based on color - This is applicable when text is represented using a different color from the other graphics in the document. The emphasis in this case is again on color segmentation/clustering as described in previous section. For example Joachim Pouderoux et al. [8] have extracted toponyms from map based on color. Second branch of research is to extract text from documents without color features. To extract text not connected to the graphics of document, Fletcher and Kasturi [9] have used connected components and some properties like Area etc. Morphology was used by Luo et al. [10]. Cao et al. [11] have proposed a method to extract text touching to the graphics, but their method is done on a vectorized image. Tombre et al. [12] have proposed a method based on connected components. P.P.Roy et al. [13] have developed a system to segment text using color, and further by using geometric features of connected components and skeleton information in each layer.

C. Recognition of digits

Handwritten digit recognition is an active topic in OCR applications and pattern classification/learning research. The literature on this topic alone is extremely huge, with a large variety of feature extraction and classification techniques being published every year. Features extracted range from geometric moments to contours and curvatures, while classification techniques range from template matching to neural networks [14]. Databases like CENPARMI [15], CEDAR [16], and MNIST [17] have been widely used in classifier design and evaluated in character recognition and classification/learning research. Every database is partitioned into a standard training data set and a test data set such that the results of different algorithms can be fairly compared. The MNIST (modified NIST) database was extracted from the NIST special databases SD3 and SD7, containing binary images of handwritten digits. SD3 and SD7 were released as the training and test data sets, respectively, for the First Census OCR Systems Conference. The database contains 60,000 handwritten digits in the training set and 10,000 handwritten digits in the test set. Different classifiers proved on this database by LeCun [18] had shown recognition rate from 88% till 99.3%

III. Topographic map Processing

Various steps in topomap processing, aiming at contour layer digitization are shown in the figure 2. The approach consists of three main modules which are Extraction, Contour line Representation and Reconstruction. In this paper, we have proposed altitude tag filtering which is highlighted in the figure 2.

As mentioned earlier, a topomap consists of various features differentiated primarily based on color. Hence clustering algorithms based on colour are applied on topomap to extract each of these features into a separate layer. Extraction module focuses on separating the contour layer from various other features of topomaps like water bodies, vegetation etc. by using Clustering. But prior to applying clustering on a topomap, the colours are quantized such that the original colors of the image are mapped to a smaller subset of colors. Clustering is followed by merging of the original clusters which further helps in extraction of meaningful layers from the topomap. The topomap is thus segmented to various layers based on their colour features, for example a layer of water bodies (in blue), a layer of text and grid lines (in black), a layer of contour lines (in brown) etc. The resulting contour layer is not clean, and consists of noise i.e some pixels of other layers not belonging to contour lines, due to problems of aliasing and false colours in a colour topographic map. Apart from contour lines and noise, the contour layer also contains the altitude values which are represented using the same color as contour lines. Hence algorithms for filtering the noise and for separation of text from graphic images become essential. After the first stage, a clean contour layer, consisting of segments of contour lines is extracted from the topomap. The filtered altitude tags form another layer and the rest of the information of topomap is stored in ‘other layer’ as it is useful in later processing steps. Each contour line from clean contour layer is extracted and represented using curve representation techniques. Similarly altitude tags are segmented and recognized. This forms the second stage of topomap processing which outputs a contour line representation and recognized altitude tags. Extraction stage leads to gaps in contour lines due to intersecting and overlapping features. So prior to vectorizing the contour layer, it becomes essential to fill the gaps. The Reconstruction module describes the algorithms for identifying contours which contribute to a gap, pairing contour segments which form a gap and then filling the gaps. This module takes a binary contour image, which is the output of Extraction module as input and generates a reconstructed binary contour image as output. The recognized elevation values are tagged to the contour lines and an attribute which stores the altitude value is updated in the

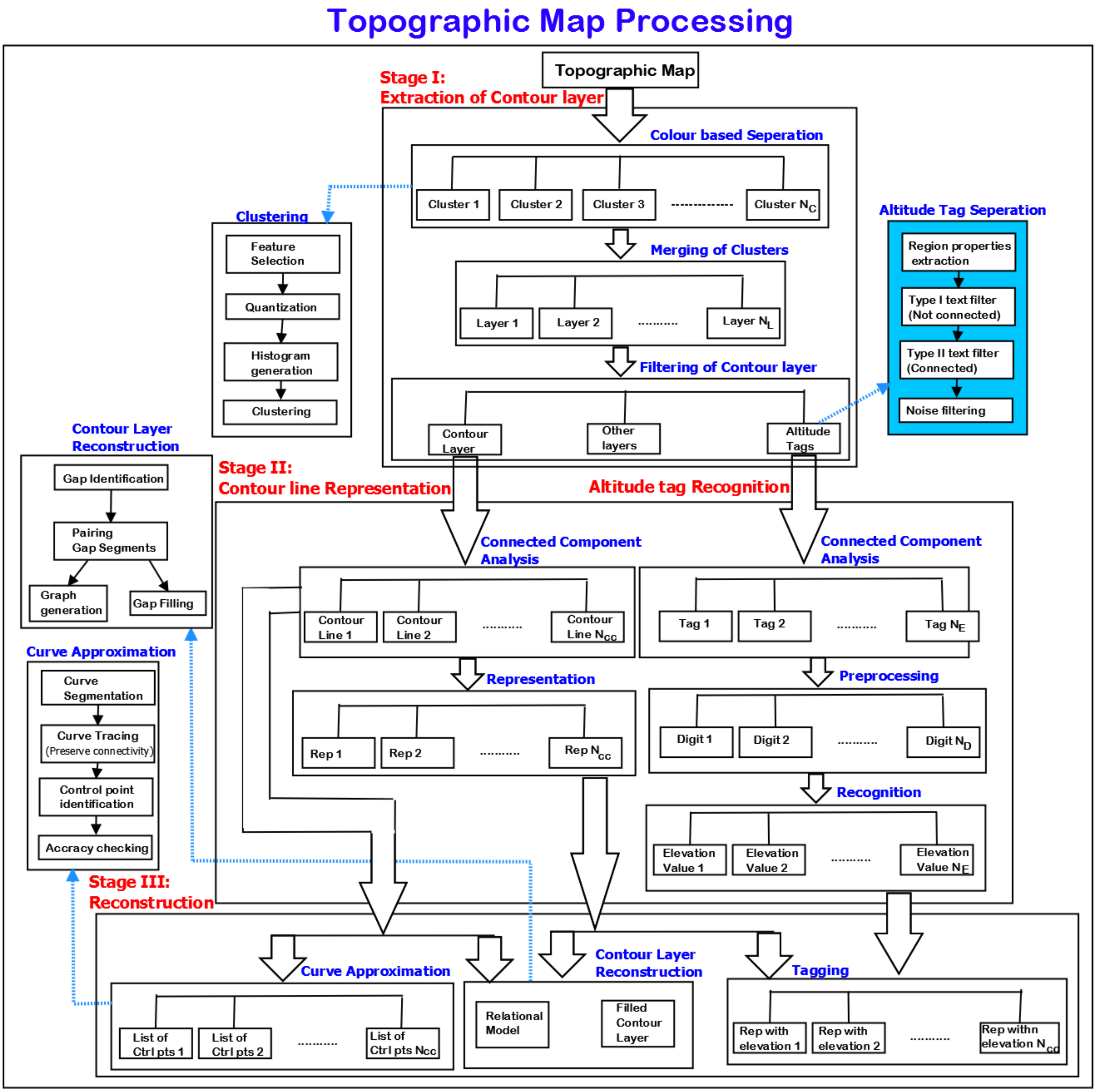
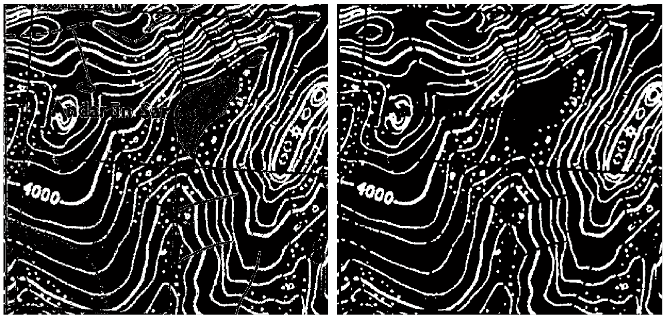
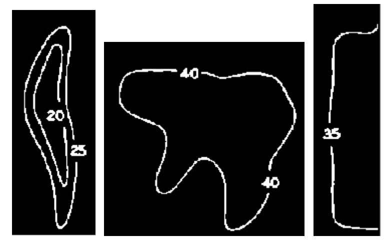


Fig. 2. Topographic map Processing



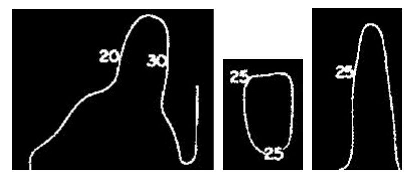
(a) Contour layer from clustering (b) Contour layer after median filter

Fig. 3. Median Filtering



(a) (b) (c)

Fig. 4. Text not connected to contour lines



(a) (b) (c)

Fig. 5. Text connected to contour lines

representation of contour line.

This paper concentrates on altitude tag extraction and recognition described in detail in the following sections.

IV. Proposed Approach : Filtering of Contour Layer

A topographic map is layered into several features based on the color. The contour layer is the salient feature of topomap, as it shows the 3D terrain in 2D. The first step of topomap segmentation has been performed using a non parametric histogram based clustering algorithm proposed by Khotanzad et al. [19] using the basic colour space, RGB. From performing clustering as mentioned, topomap is divided into large number of clusters. However each of these clusters does not represent a meaningful layer of topomap, some of them may be noise which have to be discarded, some clusters have to be merged to form a layer. Minimum distance hierarchical clustering has been employed to merge the clusters. The distances between the cluster centres is measured, and the clusters whose distance is below a certain threshold are merged. The details of extraction of contour layer are explained in [3].

However as can be seen in the figure 3(a) the contour layer extracted from a topomap is not clean. This is due to the fact that segmentation is not efficient due to various problems induced by scanning and some due to inherent characteristics of topomap. Apart from the contour lines, which connect the points of equal elevation, the contour layer also has noise, symbols and altitude tags. The text cannot be filtered in the Extraction stage as the altitude values are of the same color as contour lines, and contour layer extraction is based on color. Filtering module consists of algorithms to filter noise and extract text to generate a clean contour image. Filtering the altitude tags from contour layer and recognizing them is an important pre cursor to contour layer vectorization.

The noise can be filtered using the standard algorithms like median filter. Figure 3 shows the contour layer obtained from clustering and the contour layer after applying median filter.

However text has to be handled in a different way. Features based on spatial arrangement and intensity distributions of pixels are necessary to separate text, which are of the same color as lines from the contour layer. The steps are explained with the help of image taken from [20].

Depending on the position of altitude values, the tags can be divided into two parts

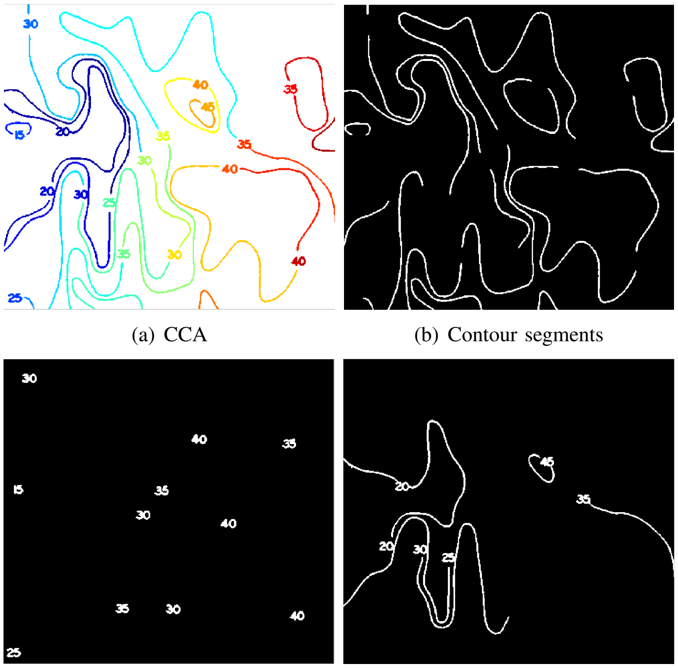
1) Text in between two contour lines, not connected to any of the contour lines, shown in figure 4.
2) Text connected to a contour line(s), shown in figure 5.

The proposed method filters text not connected to contour layers by a method proposed by Fletcher et.al. Text connected to contours is filtered using spatial arrangement and intensity distributions. Major steps in the process are:

1) Perform Connected Component Analysis (CCA) and compute properties of each component like Area of minimum bounding rectangle (MBR), Aspect ratio of MBR, total object pixels etc.
2) Cluster components based on properties extracted in step 1.
3) Filter text/symbols not connected to contours
4) Filter text connected to contour lines
5) Filter noise

A. Filter text not connected to contours

Connected component analysis & Properties Extraction: Connected component analysis/labelling (CCA) is a basic and widely used technique in image processing and pattern recognition. Connected components labelling scans an image and groups its pixels into components based on pixel connectivity (4-connected or 8-connected), i.e. all pixels in a connected component share similar pixel intensity values and are connected with each other. Once all groups have been determined, each pixel is labelled with a gray level or a color according to the component it was assigned to. Connected component labelling works by scanning an image, pixel-by-pixel (from top to bottom and left to right) in order to identify connected pixel regions, i.e. regions of adjacent pixels which share the same set of intensity values V, for a binary image V=1.



(a) CCA (b) Contour segments

(c) Altitude tags (d) Contours with attached text

Fig. 6. Classification of connected components of contour layer

When CCA is applied on the contour layer the components obtained can be classified into three categories:

1) Segments of Contour lines only
2) Altitude tags/Noise
3) Contour lines with the altitude tags attached

For example when CCA is applied on the portion of contour layer, 34 components are obtained which are shown in figure 6(a) using distinct colour to represent each component. The three categories of components are shown in figures 6(b), 6(c), 6(d) respectively.

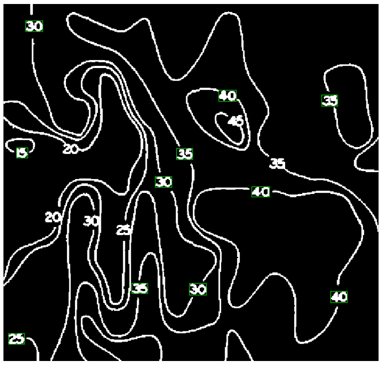
To filter the text not attached to the contours, i.e. components belonging to the second category, method proposed by Fletcher et al. [9] is employed. Each component is enclosed in a circumscribing rectangle (minimum bounding rectangle, MBR) and the region properties of the component like MBR area, aspect ratio of MBR and the total object pixels are computed. Table I lists the properties of components shown in figure 6(a)

To filter based on the properties, thresholds of each of them have to be identified. We have employed a Clustering based approach to cluster the components into two groups of symbols/text and contours. This clustering problem (2 clusters) has been attempted by one of the popular algorithms - Expectation Maximization (EM) algorithm. The EM algorithm is an iterative algorithm for calculating the maximum-likelihood or maximum-a-posteriori estimates of finite mixture models, when the observations can be viewed as incomplete data i.e. when there is a many to one mapping from underlying distribution to the distribution governing the observation [21]. Each iteration of the algorithm consists of an expectation step (E-Step) followed by a maximization step (M-Step). The E-Step is with respect to the unknown underlying variables, using the current estimate of parameters and conditioned upon the observation, the M-Step then provides new ML estimates/updates of the parameters given in the E-Step. These two steps are iterated until convergence. To use EM Algorithm, the number of component densities 'C' in the mixture has to be known and an initial guess for the value of component parameters $\Theta^{(0)} = (\mu_k^{(0)}, \sigma_k^{(0)})$ has to be made. Once initial estimates are found, the parameter estimates are updated using iterative EM update equations. We have used the pixel density (Area of MBR/Total object pixels) and aspect ratio to cluster the components into two clusters. Data is divided into two (C) equal groups, and $\mu_k^{(0)}, \sigma_k^{(0)}$ are initialized to mean and standard deviation of each of these groups respectively. After EM is run, each data point is assigned a posterior probability, based on which each component is classified as belonging to either the text or contour. From the data given in table I, components 6, 7, 8, 17, 19, 21, 23, 26, 28, 31, 33 are classified as belonging to text. Figure 7(a) shows these components enclosed in green rectangles.

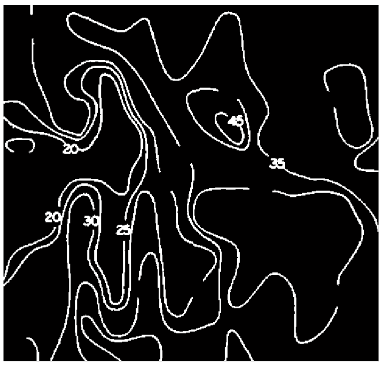
Once the text only components are filtered, altitude tags connected to contour lines have to be filtered, which is discussed in next section.

TABLE I
REGION PROPERTIES OF COMPONENTS OF CONTOUR LAYER

CC	MBR Area	Aspect Ratio	Density
1	33998	1.07	30.52
2	28380	1.04	20.11
3	4290	1.02	19.32
4	22500	1.00	15.77
5	703	1.95	4.29
6	260	1.54	1.51
7	182	1.08	1.43
8	273	1.62	1.55
9	464	1.81	5.21
10	42	4.67	1.05
11	26394	1.04	24.46
12	11385	3.76	18.45
13	45180	1.39	25.95
14	7840	1.23	13.18
15	5757	1.77	11.33
16	20410	1.21	15.27
17	260	1.54	1.53
18	21630	2.04	21.42
19	260	1.54	1.65
20	6384	1.96	17.88
21	280	1.43	1.68
22	6162	1.01	11.69
23	315	1.40	1.83
24	34547	1.08	30.25
25	1517	1.11	3.81
26	308	1.57	1.68
27	1404	1.08	6.47
28	286	1.69	1.71
29	12285	1.48	24.47
30	13334	1.04	23.98
31	280	1.43	1.68
32	6572	1.71	11.10
33	294	1.50	1.79
34	1290	1.43	6.94



(a) Components filtered



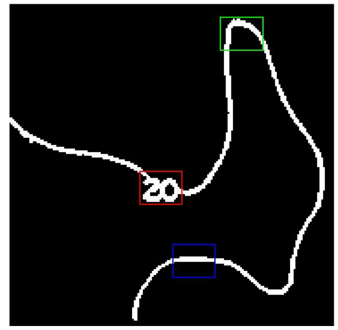
(b) Filtered Contour layer

Fig. 7. Filtering Type I text

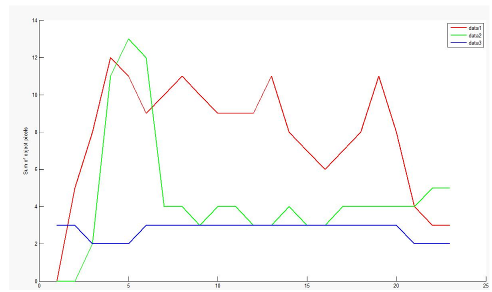
B. Filter text connected to contours

As can be seen in Figure 7(b), the contour layer still has labels which are attached to it. As region properties fail to help us classify, a more local approach taking into account the spatial characteristics of the image is essential. We have thus used row and column projections of the binary regions. Figure 8(a) shows one of the components of contour layer which has text attached. Different binary regions of the segment have been enclosed by three rectangles of same size, area enclosed in red is the text, area in green is the contour at a bend (two segments occur in the same window) and blue is a single contour segment. The row and column projections of the areas enclosed are shown in figure 8(b) and 8(c). As it can be seen from the plots, the projections vary in three cases and hence can be used to differentiate text from contours in each component. However the size of window has to be arrived at, which is done with the help of text (not connected to contours) filtered from the image.

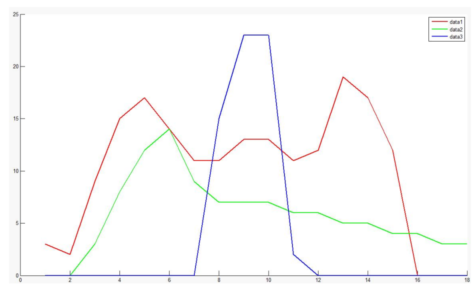
1) The properties of filtered MBRs are stored for further processing. The maximum of all the filtered MBRs dimensions is stored in w, and average pixel count is stored in T_{pc}.
2) The following procedure is followed on each component which does not belong to the text. The locations of all the object pixels in the component are stored.
3) At each object pixel, a neighbourhood of size w is considered.



(a)



(b)



(c)

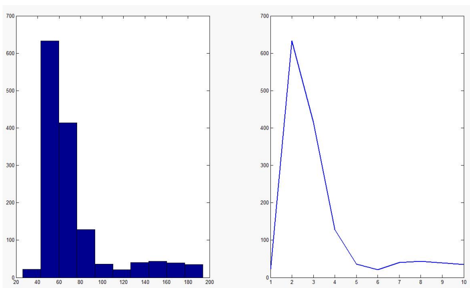
Fig. 8. Row & Column projections of regions of text and contours

4) Total count of the pixels in the window, i.e. area under the row or column projection curve, is computed. The figure 9 shows two images of contour lines with text and without text, where each pixel is replaced by the total count of pixels around a window, centred at that pixel. The image is thus in gray scale, which is termed as Projection image. The figure also shows the histogram plot of object pixel intensities of the two images.
5) It can be seen in the figure, that the pixels where text is present are brighter, and the histogram has values beyond a threshold, when compared to contour without text. Figure 10 shows the pixels whose intensity value is greater than a threshold
6) If the count is greater than threshold, T_{pc}, it belongs to the text and hence is made equal to background. (zero in this case).

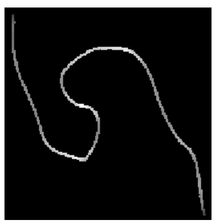
V. Recognition of altitude tags

In the previous section, filtering of altitude tags of contour lines has been explained. These altitude tags are important as they convey the elevation information of the contour line.

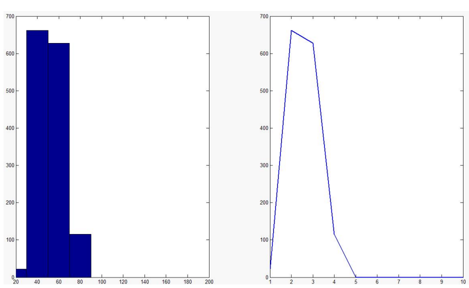
(a) Contour with text attached



(b) Histogram plot



(c) Contour without text



(d) Histogram plot

Fig. 9. Histogram plots of Projection images

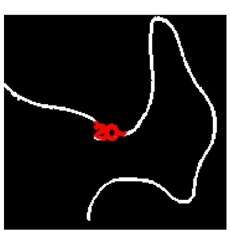


Fig. 10. Pixels with higher intensities

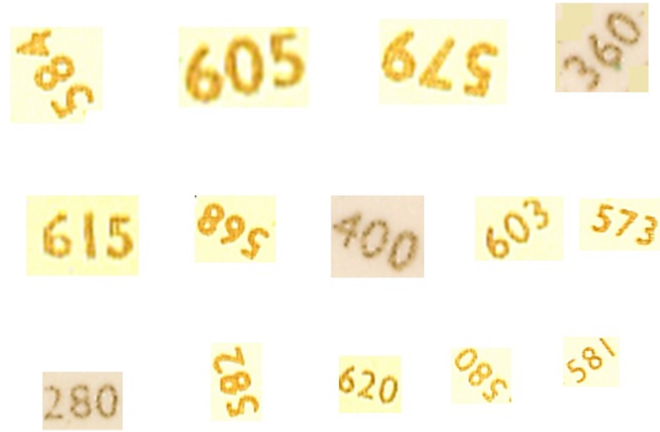


Fig. 11. Sample altitude tags of topomaps

To obtain a three-dimensional surface from the contour layer, the altitude tag values have to be recognized and tagged to the contour lines. Altitude tag recognition is discussed in this section. Figure 11 shows the samples of altitude tags extracted from a topomap.

The following properties of tags in the topomap make altitude recognition a challenging problem:

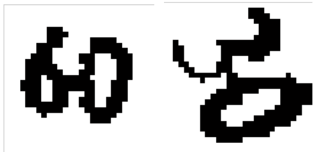
1) Altitude tags do not belong to any standard font, the tags are not printed but hand written using markers.
2) Altitude tags are not aligned along any axis, but rotated along the contour line to which the value has to be tagged.
3) Altitude tags are noisy. As text is extracted from a scanned colour topographic map, gaps are formed due to variations in colour introduced by the scanning process. Also when text attached to contour lines is filtered from the contour layer, some noise gets attached to the tag.

The altitude tag recognition is divided into three stages:

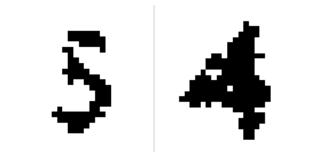
1) Stage I : Preprocessing
   - Separation of digits in the altitude tag
   - Size normalization
   - Rotation normalization
2) Stage II: Recognition of digits
3) Stage III: Altitude tag recognition
   - Recognition of the whole altitude tag
   - Validation

A. Preprocessing

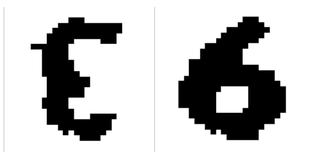
1) Connected component analysis(CCA) is applied on each altitude tag to separate the digits. However this does not help when the digits in the tag are connected as shown in the figure 12(a). In the case of topographic maps, it can be assumed that the number of digits in the altitude tags across the map is same. Hence region properties of the tags can be used to segment the digits. When CCA does not yield the same number of components as the digits in an altitude tag, properties of each component like height,width are measured and and the component is split across the major dimension.



(a) Connected digits



(b) Noise/Holes in digits



(c) Improper rotation

Fig. 12. Problems with extracting digits from altitude tag

Fig. 13. Sample digits obtained after pre processing

2) Morphological closing operation is performed to fill any holes in the digit as shown in figure 12(b)
3) Each digit after segmenting, has to be rotated, so that all the digits are aligned along a common axis. Principal axis rotation has been used, i.e. the image is rotated along the first principal component axis, i.e along which the variance is maximum. The spatial co-ordinates of object pixels are found, principal components are computed, and then image is rotated at an angle of Cos^{-1} of first principal component. However the digits may be rotated by 180 deg, as shown in figure 12(c), and 6 and 9 are incorrectly recognized.
4) The digit is normalized to a standard size

Figure 13 shows some digits obtained after the preprocessing on the altitude tags.

Fig. 14. Sample images of MNIST data

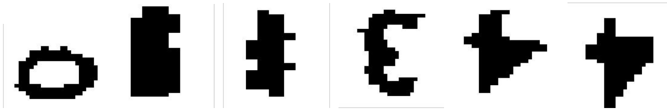
B. Recognition of digits

The convolutional neural network created by LeCun et.al [18] has been used for the recognition of altitude tags. This class of networks have been successfully used in many practical applications, such as handwritten digits recognition, face detection, robot navigation and others. This network provides some degree of shift, scale and distortion invariance as their basic architectural ideas include local receptive fields, shared weights (or weight replication), and spatial or temporal sub sampling.

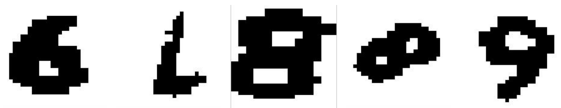
Topology of LeNet - 5 [18]: The architecture of LeNet - 5, which is a typical convolutional neural network used for recognizing characters, is shown in figure 15. LeNet-5 consists of eight layers. The input layer size is 32 X 32, even though the input images used are at most 20 X 20, so that the relevant features are then guaranteed to be contained in all feature maps and not get lost because they are near the boundary. The first convolutional layer has six feature maps, each of which has a resolution of 28 X 28, with a receptive field of 5 X 5. The second layer, or the first subsampling layer, contains six feature maps of size 14 X 14. The third layer is another convolutional layer and has 16 feature maps with size 10 X 10, with a receptive field of 5 X 5. The fourth layer contains 16 feature maps as well, each of which is of size 5 X 5. The fifth layer is a convolutional layer with 120 feature maps, again with a receptive field of 5 X 5. Then follows a layer with 84 neurons, which is fully interconnected with the previous layer. All neurons up to and including the sixth layer compute their input by calculating the weighted sum and feeding the result to the squashing function

$$f(u) = A \ \tanh(Su) \tag{1}$$

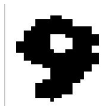
where A was chosen to be 1.7159, and S determines the slope of the function at the origin. Finally, the last layer contains ten radial basis function neurons. The weights of the output



(a) O as 2 (b) 1 as 5 (c) 1 as 8 (d) 3 as 8 (e) 4 as 7 (f) 4 as 9



(g) 6 as 5 (h) 7 as 1 (i) 8 as 5 (j) 8 as 9 (k) 9 as 5



(l) 9 as 8

Fig. 16. Digits Misclassified

layer were initialized by hand (as opposed to random for all other weights) to map to a bitmap representation of the ASCII character set, in order to overcome difficulties with characters that are similar, and thus give a similar output.

The convolutional neural network is trained using MNIST database. Some images of MNIST database are shown in Figure 14. The digits obtained from the previous step are normalized such that the mean is 0 and variance is 1 . When tested on digits extracted from altitude tags of some topographic maps, the accuracy is about 73%. Of the 106 digits, 29 have been misclassified, some of which are shown in the figure 16. It can be noted that most of the errors in recognition are due to presence of noise or orientation of the digits.

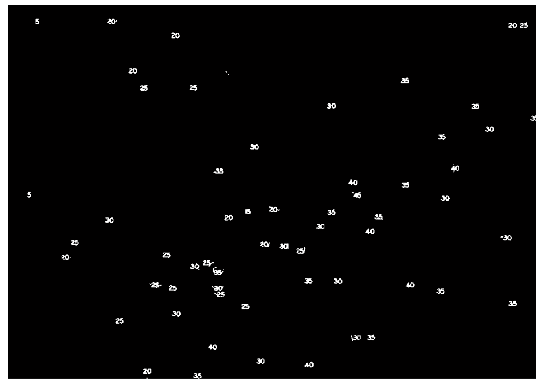
C. Altitude tag recognition

The digits segmented from the set of altitude tags are recognized using LeNet - 5. The digits corresponding to a tag are grouped and the value of the whole tag is computed based on the place value of the digit. However, the accuracy of recognition of digits is not 100%, and a post processing stage is essential to validate the recognition. The error rate of recognition can be reduced by adding knowledge to the system. The following rules of altitude tags are used in validating the recognized tags:

1) Altitude values are tagged to contour lines through out the topomap. Hence, same contour line has more than one altitude tag at different places. Altitude values tagged to the same contour line have the same value.
2) Altitude values change in the intervals of 5 or 10 across the contour lines.
3) Altitude values of the intermediate contour lines which lie between two index contour lines lie in the range of the altitude values of the index lines.

The Validation stage has not been included in the tool at this stage as it demands solutions to several issues regarding knowledge representation and altitude value tagging to contour lines. This forms our future work.

(a) Image from [20]



(b)

Fig. 17. Altitude tag filtering

VI. Results & Conclusion

In the contour layer, shown in Figure 17, altitude values which belong to the first category, are filtered using properties of minimum bounding rectangle. The other altitude tags are selected using projection images of connected components. Digits are extracted from the separated altitude tags and given as input to LeNet - 5 network. Of the total 112 digits, 61 have been correctly recognized. Of the 51 mis-classified digits, 19 errors are due to improper pre processing like improper segmentation of digits from the tag which leads to presence of noise or holes in the digit. All the fours (4) have not been recognized as they vary to a greater degree from the standard training set of the network shown in figure 15. Similarly 2's also have not been recognized at all. Hence from the remaining 93 digits, 32 digits have been wrongly recognized which amounts to an accuracy of about 66%. Table II shows the recognition of individual digits. As mentioned earlier this can be improved by adding knowledge about the system and by training the network using digits from the altitude tags of various topographic maps, which is out of scope of this paper. For example, a contour line is tagged with a value of 30 at ten different places. However, out of 10 tags, two digits of a tag have been correctly recognized in 6 tags. The recognition errors in the remaining four tags can be corrected as we know that they belong to the same contour as the correctly recognized tag 30.

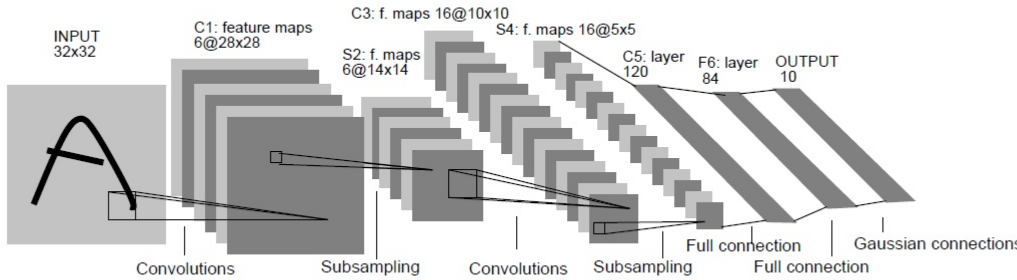


Fig. 15. Architecture of LeNet - 5 [18]

TABLE II
ALTITUDE DIGIT RECOGNITION

Digit	Total No	Errors due to Preprocessing	Errors in Recognition	Accuracy %
0	28	1	3	88
1	2	2	0	0
2	20	9	11	0
3	27	2	1	96
4	7	3	4	0
5	28	2	13	50

A colour topographic map is clustered into layers like contour, vegetation, water bodies etc based on colour features. However the contour layer is subjected to further processing as the altitude tags which represent the elevation value of the contour are also present along with the lines. An approach to filter altitude tags attached to the contours is discussed. It can be observed that the labels thus extracted can be further useful to tag the contour lines with the altitude values and have a 3D model. The recognition of altitude tags using LeNet - 5 has been discussed. All the steps in the proposed approach are highly amenable to complex shapes of contour lines. Also the process is scalable for 3D model generation of contour layer, which forms our future work.

REFERENCES

[1] C. Albert and K.W.Yeung, *Concepts and techniques of Geographic Information Systems*. India: PH Series in Geographic Information Science, Prentice-Hall.

[2] A. Khotanzad and E. Zink, "Contour line and geographic feature extraction from usgs color topographical paper maps," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, 2003.

[3] B.Sandhya, A. Agarwal, C. Rao, and R. Wankar, "Automatic gap identification towards efficient contour line reconstruction in topographic maps," *Asia Modelling Symposium*, 2009.

[4] M. Ansoult, P. Soille, and J. Loodts, "Mathematical morphology: A toolfor automated gis data acquisition from scanned thematic maps," *Photogramm.Eng. Remote Sens*, vol. 56, no. 9, p. 1263 1271, 1990.

[5] N. Ebi, B. Lauterbach, and W. Anheier, "An image analysis system for automatic data acquisition from colored scanned maps," *Machine Vision Applications*, vol. 7, no. 3, p. 148 164, 1994.

[6] Y. Chen, R. Wang, and J. Qian, "Extracting contour lines from common-conditioned topographic maps," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 4, April 2006.

[7] C. Rids, C. Rusu, and J. Astolu, "Contours extraction from a colour coded relief scanned map," *International Conference on Image Processing (ICIP)*, 2004.

[8] J. Pouderoux, J. C. Gonzato, A. Pereira, and P. Guitton, "Toponym recognition in scanned color topographic maps," *International Conference on Document Analysis and Recognition(ICDAR)*, pp. 531 – 535, September 2007.

[9] L. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Transactions on pattern Analysis and Machine Intelligence*, vol. 10, pp. 910 – 918, 1988.

[10] H.Z.Luo, G.Agam, and I. Dinstein, "Directional mathematical morphology approach for line thinning and extraction of character strings from maps and line drawings," *International Conference on Document Analysis and Recognition(ICDAR)*, vol. 1, p. 257, 1995.

[11] R.Cao and C.L.Tan, "Text/graphics separation in maps," *Proceedings of 4th IAPR International Workshop on Graphics Recognition*, vol. 1, p. 44 48, September 2001.

[12] K.Tombre, S.Tabbone, L.Peissier, B.Lamiroy, and P.Dosch, "Text /graphics separation revisited." *In Document Analysis Systems V*, vol. 2423, pp. 615 – 620, 2002.

[13] P.P.Roy, E. Vazquez, J. Llados, R. Baldrich, and U. Pal, "A system to segment text and symbols from color maps," *7th International Workshop on Graphics Recognition Recent Advances and New Opportunities: LNCS*, vol. 5046, pp. 245 – 256, 2008.

[14] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques," *Pattern Recognition*, vol. 36, pp. 2271 – 2285, 2003.

[15] C. Suen, "Computer recognition of unconstrained handwritten numerals," *Proc.IEEE*, vol. 80, no. 7, pp. 1162 – 1180, 1992.

[16] D. Lee and S. Srihari, "Hand printed digit recognition: a comparison of algorithms," *Proceedings of the Third International Workshop on Frontiers of Handwriting Recognition*, pp. 153 – 164, 1993.

[17] Y.LeCun, "Comparison of learning algorithms for handwritten digit recognition," *F. Fogelman-Soulie,P.Gallinari (Eds.), Proceedings of the International Conference on Artificial Neural Networks*, pp. 53 – 60, 1995.

[18] Y.LeCun, L.Bottou, Y.Bengio, and P.Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278 – 2344, November 1998.

[19] A. Khotanzad and A. Bouarfa, "Image segmentation by a parallel, non-parametric histogram based clustering algorithm," *Pattern Recognition*, vol. 23, no. 9, pp. 961 – 973, 1990.

[20] J. Pouderoux and S. Spinello, "Global contour lines reconstruction in topographic maps," *Ninth International Conference on Document Analysis and Recognition,ICDAR*, pp. 779 – 783, 2007.

[21] A.P.Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Statistics Society*, vol. 39, no. 1, pp. 1–38, 1977.