MVA'90 IAPR Workshop on Machine Vision Applications Nov. 28-30,1990, Tokyo

AUTOMATIC DIGITIZING OF THE COLOUR-LAYER OF THEMATIC MAPS

Rune Espelid, Nils Arne Alvsvaag, Jan Eileng, Ivar Skauge

IBM Bergen Scientific Centre
Thormohlensgate 55, 5008 BERGEN
Norway

Abstract

A method for automatic classification and description of the colour-layer of thematic maps is presented. Owing to the type of noise appearing in the scanned images, we find it useful to combine pixel classification and raster-to-vector conversion in order to produce acceptable results. The colour value of an area is determined from a selectively sampled set of pixels and a set of threshold values. Possible borderlines are produced by vectorizing the line-network of the image, and false borderlines are removed by comparing neighbouring sub-areas.

Experimental results are discussed, and some test maps are presented to illustrate the performance of the system.

INTRODUCTION

A common method for representing thematic information in cartography is to use colour maps. Such maps may show distribution of soil types, vegetation cover, mineral resources, human population and so forth.

In this paper we present a method for automatic classification and description of the coloured areas of scanned maps.

Two reasons are apparent why it is of interest to digitize such information: firstly, to bring it into an information database; secondly, to compress the volume of data for storing scanned maps. To illustrate the last problem, consider a 40 by 25 inches wide sheet of map, scanned at 150 dpi with 16 bpp. The resulting file size will be around 30 Mbytes.

A number of challenging situations may occur in typical maps. In border regions colours may overlap, expanding one area beyond its border, or resulting in undefined colours. The colours printed on the map are represented by a raster pattern, implying that when scanned at a reasonable resolution, a high percentage of the pixels are given a value corresponding to the background of the paper, rather than the thematic colour. Furthermore, some colours representing different areas, may in certain regions appear to be very similar, e.g. caused by overlaid colour information.

These conditions make a pixel filtering and classification procedure insufficient to produce an accurate description of the borderlines. On the other hand, a solution on the basis of a pure raster-to-vector conversion process is made impossible since the maps may contain a mixture of lines (representing roads, railroads, rivers, grid, etc.), as well as text and symbols, in addition to the borderlines of the coloured areas. The coloured areas are therefore usually split into a number of smaller areas by the line network.

Our procedure combines the two methods of pixel classification and raster-to-vector conversion into an algorithm which is robust to most of the noise and distortions commonly found. The procedure has one limitation: it requires that the borders of the coloured areas of interest are defined by lines, clearly distinguishable from the areas themselves.

THE COMBINED APPROACH

A summary of the processing procedure which has been used is presented below. The raster-to-vector conversion system used is described in more detail by Espelid and Eileng in [1]

- Reduce the volume of data by thresholding the image. All pixels are classified into the following categories: black (line network including the borderlines); a number of colours corresponding to the number of area types; and white representing undefined pixel values.
- Copy the black pixels into a separate bitmap which is then processed by a raster-to-vector (r-t-v) conversion subsystem. (See figure 1 for an example of a bitmap image.) The r-t-v process includes the following steps:
  - Produce a chain coded description of all contours.
  - Delete all internal contours smaller than a critical size.
  - Apply a contour-based thinning procedure to the remaining contours.
  - Produce a mathematical graph description of the resulting skeleton with nodes representing critical points (end-points and junction-points), and edges representing the connected chains between the critical points. Each loop in this graph describes a sub-area in the map.
- For each loop in the graph, determine its colour value by evaluating the colour of the corresponding sub-area. Pixel values are sampled from the thresholded image.
- Compare all neighbouring loops of the mathematical graph, and delete edges which separate loops with the same colour value. The resulting loops describe the areas of interest.
- The chains describing the resulting loops are exactly the borderline we are interested in. The borderlines are polygonized and saved, together with area-code information.

Figure 1. An example of a bitmap image (corresponding to figure 3 a)), constructed from all pixels classified as black.

THRESHOLDING THE SCANNED MAP

It is assumed that the threshold values can be determined once for a large batch of maps. An interactive procedure, including some test scanning, may turn this task into a few minutes job for the operator. We have not tested any automatic methods for segmentation of the coloured images.

Even though the practical benefit of automatic colour segmentation is limited in our application, it is an interesting problem. For further study the articles written by Otha and Wright ([2] and [3]) should be consulted.

Thresholding the image as the first step is motivated by the assumption that the amount of data should be reduced as much as possible before being transferred from the scanner to the computer.

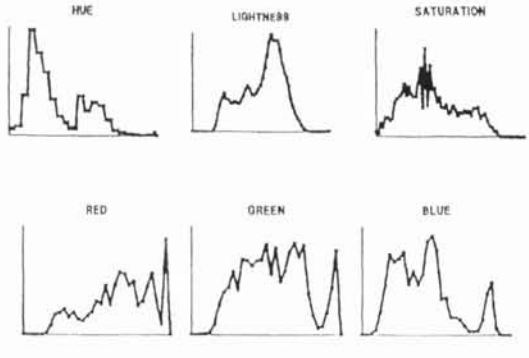

Figure 2. Colour histograms produced from the test map shown in figure 3 a).

THE THINNING ALGORITHM

The thinning algorithm used is important for the efficiency and quality of the total procedure, we find it appropriate to give a brief explanation of how it works. A complete description can be found in [4]

The thinning algorithm accepts as input a set of chain-coded contours describing the object to be thinned. During one iteration a new set of contours is created inside the existing ones. This is done by traversing each contour sequentially, and for each position on the contour, generate a set of new chain elements on its object side. When two chain sections pass through the same position, an element of the skeleton has been detected. This process is iterated until the original contours are reduced to a chain-code describing the skeleton of the object.

NOISE REDUCTION

The specified algorithm has a number of noise reduction effects, which makes the procedure quite robust and general.

Deleting small sub-areas

The contour tracing routine produces closed chain-coded contours from all object boundaries in the image, including all internal boundaries. As an example, one single white pixel inside a black area will in principle cause a closed contour to be produced.

In some maps there may exist symbols with small internal areas which have no defined colour. In some of our test examples, the railroad line font is a thick line with internal white dots. We therefore delete all contours smaller than or equal to the size of these small internal areas.

The consequence of deleting an internal contour is identical to painting black the corresponding subscribed region. This area will then be treated by the subsequent thinning.

Deleting text and line ends

The map may contain text and other symbols printed in black, or which after thresholding have been classified as black. In our test maps parts of the dark brown elevation contour lines were taken for black.

Figure 3. a) Original image. b) Thresholded image. c) Classified areas. d) Borderlines of classified areas.

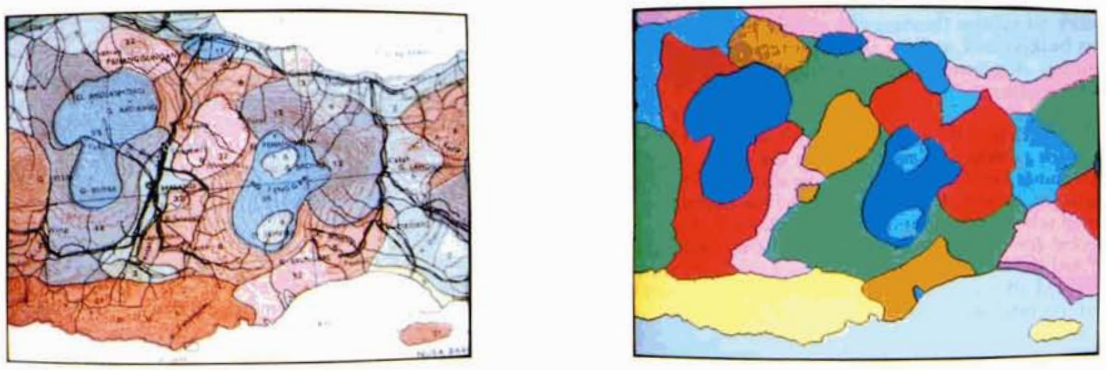

Figure 4. a) Original image. b) Classified image.

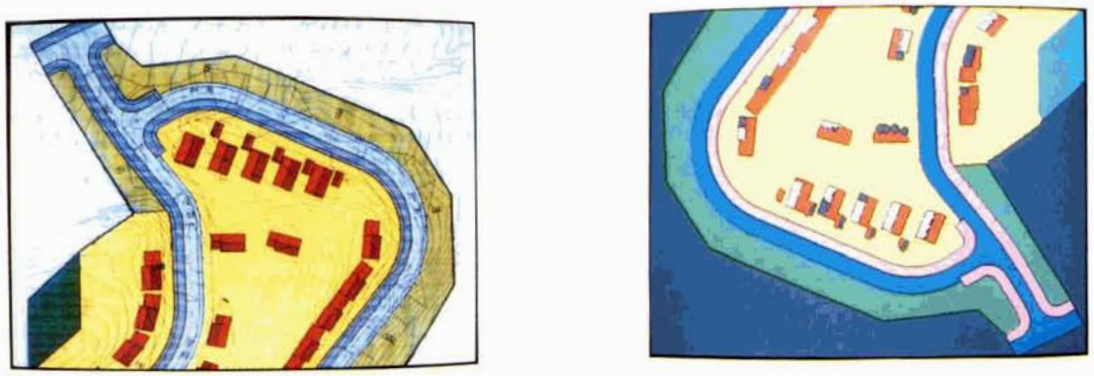

Figure 5. a) Original image. b) Classified image.

We assume that the borderlines of the coloured areas make a single connected network of lines, with no unattached line ends; and we also assume that the network incorporating the borderlines is the largest network in the map. Then we can obviously consider all smaller stand alone networks, or components, as being irrelevant for our purpose and delete them. This would then for example remove much of the text.

For the same reason we can also consider all lines, connected to the main network but with an unattached end, as irrelevant and candidates for being deleted.

Optimal handling of undefined areas

The black line pattern may cause regions in the map to be undefined. The more layers of information the map contains, the larger these regions may be. As an example, a city having many connecting lines and symbols overlaid may be positioned on the border between two regions of different soil type. Thick railroad lines and wide rivers passing through may cause a significant area to have an undefined soil type.

Our algorithm handles all undefined areas in the same way. The skeleton of these areas, produced by the thinning operation, is taken as the border line between the adjacent areas.

Selective sampling when determining the colour value

The thresholded image usually contains a considerable amount of noise, or undefined pixel values. The thresholding may cause many pixels to be misclassified, especially when different colours appear very similar. In border regions, for instance, where colours may overlap, misclassifications occur frequently. The colour value of each sub-area is therefore not directly available in the thresholded image.

We determine the colour value of each sub-area by counting pixel values in the internal region of the sub-area and allowing the majority to win. The internal region is obtained by taking the original chain-coded contour description of the actual sub-area, and doing an inverse thinning (we simply redefine the line network as background, and the areas as foreground). The number of thinning iterations reflects the diameter of the uncertain boundary regions.

Once the internal of the region is reached, a number of sampling schemes are possible: to sample along the chain only; to continue to iterate the thinning and sample along the chain between iterations; or to sample all pixels inside the internal region. A combined scheme which optimizes correctness and speed should be selected for each class of maps. In the test example below we iterated the thinning eight times (from the original position), and sampled all pixels along the chain between the last four iterations, no pixel was selected more than once.

EXPERIMENTAL RESULTS

A prototype of the automatic digitizing system has been implemented in C and installed and executed on an IBM Aix workstation. The maps have been scanned using a Howtech colour scanner. The resolution produced was 150 dpi with 5 bits per RGB-colour.

Figures 3 to 5 show different map examples which have been processed.

The system handles situations with partly overlapping, similar or poor quality colours, and it is almost undisturbed by overlying black lines representing various themes.

The system very seldom looses any area. Loss of area can occur when gaps in the borderlines have not been closed, and when colours of different areas appear very similar. False borders are maintained in rare cases. When it happens, most often a human observer also would have problems.

A reasonably dense polygonization of the borderlines implies that the original image can be compressed with a factor of 1:400.

ENHANCEMENTS OF OUR ALGORITHM

A few obvious enhancements to the current implementation should be mentioned.

The network of lines defining the borders of the coloured areas may, due to various reasons, have gaps which split the borderlines into segments. To be prepared for such situations we propose that a gap closing procedure is applied to the binary image containing the line-network. This could be based on simple low level image operations (SHIFT and OR). However, we would rather also take advantage of the fact that all loose ends can easily be identified in the graph, and on the basis of their features (position, direction, thickness) design a more sophisticated gap closing scheme.

Instead of considering all small sub-areas as noise and deleting them, it would be better firstly to do pixel value sampling inside them and then delete only those which do not contain a defined colour value.

The noise reduction functions could be tailord and extended for single applications, and thereby increase the reliability of the systems.

CONCLUSION

This paper has outlined a combined approach, based on pixel classification and raster-to-vector conversion, to the problem of digitizing automatically certain types of coloured maps. The procedure appears to be robust to various types of noise, and the performance in terms of correct area classification is high.

References

[1] Espelid, R., J.A. Eileng. *A raster-to-vector conversion system producing high quality geometric entities.* Proceedings of The 6th Scandinavian Conference on Image Analysis, June 1989, Oulu, Finland.

[2] Ohta,Y. *Knowledge-based interpretation of outdoor natural colour scenes.* Research notes in Artificial Intelligence 4, Pitman, UK, 1985

[3] Wright, W.A. *A Markov random field approach to data fusion and colour segmentation.* Image and Vision Computing, vol. 7, no. 2, May 1989.

[4] Kwock, P.A. *A Thinning algorithm by contour generation.* Communications of the ACM, vol. 31, November 1988.