

ELEVATION CONTOURS EXTRACTION FROM A COLOR-CODED RELIEF SCANNED MAP

Carol Rus¹, Corneliu Rusu^{1,2}, and Jaakko Astola²

¹Dept. of Electronics and Telecommunications, Technical University of Cluj-Napoca, ROMANIA
Email: Carol.Rus@bel.utcluj.ro; Corneliu.Rusu@bel.utcluj.ro

²Tampere International Center for Signal Processing, Tampere University of Technology, FINLAND
Email: Corneliu.Rusu@tut.fi; Jaakko.Astola@tut.fi

ABSTRACT

The focus of this paper is in extracting the elevation contours from a color-coded relief map. The map is preprocessed using a WHMM (wavelet domain hidden Markov model) based algorithm for denoising. Then a linework removal algorithm – that uses a vector angle edge detector with a saturation-based combination of hue and intensity planes – is applied. Each distinct elevation region must be solid colored using distinct colors. In order to achieve this, for the printed scanned map, we used a vector median filtering, and a color clustering process – based on a minimum variance quantization algorithm. The gap filling problem can be solved with a multi-pass majority filter algorithm. The elevation contours are extracted by detecting the borders between colors through a sliding-neighborhood operation.

1. INTRODUCTION

Digital elevation models (DEMs) are bidimensional arrays of values representing the spatial distribution of the altitude data corresponding to a terrain area. Getting high quality DEM data for some locations on the earth is easy (e.g. US DEMs are available at USGS). However, this problem can be more challenging for many other earth surface areas. A low cost solution for producing a DEM consists in interpolating the elevation values extracted from a scanned map.

A widely used approach for producing digital elevation models from scanned map is the derivation of DEMs from topographical contour maps [1] [2]. Color-coded maps represent, also, a way to express the altitude information. Most of the printed GIS maps combine different layers [3]. So before we can have the altitude information, we must remove all the data belonging to non-elevation layers. Thin and sharp enough edges within a map are referred to as linework. Within color-coded relief maps, in generally, the unwanted information is linework, while within topographical contour maps the altitude data is a specific colored linework.

In this paper we propose an approach for extracting the elevation contours from a color-coded relief map. The paper is organized as follows. A preprocessing method using a WHMM (wavelet domain hidden Markov model) algorithm for denoising [4, 5] is presented in Section 2. In the next Section, a linework removal algorithm – that uses a vector angle edge detector with a saturation-based combination of hue and intensity planes [6] – is applied. Section 4 is concerned with the color clustering problem. Each distinct elevation region must be solid colored using distinct colors. In order to achieve this, for the printed scanned map, we used a vector median filtering [7], and a color clustering process – based on a minimum variance quantization algorithm [8] [9]. Eliminating the unwanted information yields in unfilled gaps within the map image. In Section 5 the gap filling problem is solved using the Wise's algorithm [3]. The resulting image contains altitude data expressed using solid colored blocks. In Section 6 we present the extraction of the elevation contours by detecting the borders between colors through a 3 by 3 window sliding-neighborhood operation.

2. WHMM DENOISING (PRE-PROCESSING)

Dithering effect [10] is widely used for printing color-coded relief maps. Dithering is a particular *noise like* technique in color-half-toning, the actual method to simulate colors with fewer colors; therefore a noise removal filtering can be applied to the map image in order to eliminate it. An answer to this problem is to apply a median filtering on the image. But the median filter does not preserve well thin edges. Because an edge detection algorithm will be applied to the map image in order to identify linework, in the next step, a more suitable solution as a preprocessing would be the WHMM noise removal method.

The noise removal algorithm based on WHMM was developed (and tested on gray level images, with white Gaussian noise added) by Crouse et al. in [5]. In this work we adapted the algorithm for a color image by applying it to

each of the RGB planes. The main drawback of the approach is the computation time.

The (wavelet domain) hidden Markov model consists in four parameters: *mu* (means matrix), *si* (variances matrix), *ps* (probability mass functions matrix), and *es* (state transition probability quadrants matrix). The model is obtained using a training step that applies the EM (expectation - maximization) algorithm, having the original image as the observation data. A WHMM models the wavelet coefficients of an image and uses two state variables: *S* (small variance), and *L* (large variance). We assumed that the wavelet coefficients have similar statistical features within the same scale. The likelihood computation step takes as inputs: the model, and the image to be denoised and gives the *pdfs* (probability density functions) for both the *S*, and the *L* plane, for each wavelet coefficient of the original image. The model, the pdf matrix and the original image wavelet coefficients are used for 2D Wiener type filtering considering only the finest scale. Larger wavelet filters are more appropriate for the noise removal. We used *Daubechies 8* wavelet.

By processing each RGB component separately with the WHMM noise removal algorithm, color distortions may appear. For the tested images the color distortions proved to be sufficiently small, so that the color clustering algorithm (that uses prior knowledge on the picture) was not affected.

3. LINEWORK REMOVAL

Linework can be viewed as edges in a scanned map image. Most of the linework are thin enough, and sharp enough edges, thus an edge detection algorithm can identify them. More edge detectors have been tested. WHMMs are particularly well suited to images containing singularities (especially when using *Haar* wavelet) [4]. We applied the algorithm on an intensity plane of the image obtained by taking a simple average of the RGB components. A well-known edge detector in color image processing is the difference vector edge detector [6] which is a 3×3 operator calculating the maximum gradient across the central pixel. The vector gradient edge detector is a local operator that computes the maximum distance in the desired metric between the center pixel and the 8-connected pixels adjacent to it. The methods were applied on the RGB map image using the Euclidean distance version and the vector angle version. The Euclidean distance is efficient in the intensity domain, while vector angle is efficient in the hue domain. Thus the methods can be combined using a trade-off parameter calculated based on the saturation or the intensity.

The best results for the tested maps were obtained with the vector angle edge detector with a saturation-based com-

bination of hue and intensity planes, denoted by C_{GV} .

$$C_{GV} = \max_{i=1..8} \{ \sqrt{\alpha(S_i) \cdot \alpha(S_0)} \cdot VA(\vec{v}_i, \vec{v}_0) + (1 - \sqrt{\alpha(S_i) \cdot \alpha(S_0)}) \cdot ED(\vec{v}_i, \vec{v}_0) \} \quad (1)$$

where $i=0$ is the central pixel, and $i=1 \dots 8$ are the neighboring pixels in 3×3 window.

The trade-off parameter α is defined by:

$$\alpha(S_i) = \frac{1}{1 + e^{-75 \cdot (S_i - 0.25)}} \quad (2)$$

where S_i is the saturation of the pixel i . The saturation is that aspect of perception that varies most strongly as more and more white light is added to a monochromatic light [9]. As the value of saturation increases, the corresponding colors (hues) vary from unsaturated (shades of gray) to fully saturated (no white component).

The following relation gives the *Euclidean distance*:

$$ED(\vec{v}_1, \vec{v}_2) = \|\vec{v}_1 - \vec{v}_2\| = \sqrt{(v_{1,1} - v_{2,1})^2 + (v_{1,2} - v_{2,2})^2 + (v_{1,3} - v_{2,3})^2} \quad (3)$$

where $\vec{v}_i = [v_{i,1} \ v_{i,2} \ v_{i,3}]^T$ is a color triplet.

Vector angle is an alternate metrics and is defined in the next equation:

$$VA(\vec{v}_1, \vec{v}_2) = \sqrt{1 - \left(\frac{\vec{v}_1^T \cdot \vec{v}_2}{\|\vec{v}_1\| \cdot \|\vec{v}_2\|} \right)^2} \quad (4)$$

The image yielded by the edge detector algorithm is then thresholded using a relative strength of maximum 35%. Figure 1-b presents a mask obtained using a threshold of 25%. The threshold value was selected in order to obtain thick enough edges, while losing less possible elevation data. The thresholding process gives a mask image for the linework within the map. The mask is a binary image that shows edges using 0 (black) pixels.

Finally, the map image is overlapped with the mask, and the resulting image will contain white pixels to indicate the linework removal.

4. COLOR CLUSTERING

The white gaps within the map after linework removal can be filled using the Wise's algorithm [3] (See next Section). But this method works well, only, with solid colored blocks. Thus we need to have a distinct color for each distinct elevation area that appears in the image. The WHMM preprocessing step has removed the dithered effect partially making easier the job for the edge detection algorithm. At this moment it is possible to apply a *vector median* filtering - based on the Euclidean distance, because there is no more

linework to detect. The purpose of the 3×3 window median filtering is to remove the remained isolated linework pixels and, also, to prepare the picture for the color-clustering algorithm.

The *minimum variance quantization* algorithm is an efficient way for clustering colors [8]. Unfortunately, before we can apply the algorithm, we need to edit the image map manually. On the tested maps the problem appeared with larger lakes. They cannot be fully eliminated by the linework removal algorithm, so the resulting image presents some blue isles surrounded by white pixels. They can be easily eliminated using manually editing. The blue spots have high hue and intensity similarity with some green spots that denotes elevation between 0 and 100 m. So the automated process cannot be implemented so easy. On the other hand they need to be removed in order to reduce distortion in the clustering process.

The minimum variance quantization algorithm uses prior knowledge on the picture; it needs as an input the number of different elevation regions within the image plus one (the white linework pixels). The specified number determines the number of optimally boxes into which the RGB color cube is divided. The algorithm works by associating pixels into groups based on the variance between their pixel values. A set of pixels may be grouped together because their distance is smaller than a specific value from the center pixel of the group. The boxes that divide the color cube vary in size, and do not necessarily cover the entire color cube. The quantization allocates more of the colormap entries to colors that appear frequently in the input image and fewer entries to colors that appear infrequently [8].

The color clustering algorithm worked well with the maps we tested. However due to some printing errors and the incompletely removal of the dithered effect some isolated wrong classified pixels appeared. They needed to be removed manually. Figure 1-c presents experimental results on the color clustering problem.

5. GAP FILLING

In order to fill in the gaps left by the linework removal, a multi-pass majority filter – the *Wise's algorithm* – [3] can be run on the color-clustered image to reclassify the white pixels on the basis of their surrounding pixels. The filter functions by taking a 3 pixel wide processing window, and then testing whether the central pixel of the window is a white pixel (linework pixel). The linework pixel located in the center of the analyzing window is then replaced by the most abundant, non-linework, colored pixel value from the surrounding 3×3 window. For large areas of white pixels, multiple passes of the algorithm are needed. For the example presented in Figure 1-d we run the algorithm 11 times in order to remove all the white pixels. The algorithm

behaved well considering that the image contained a lot of unknown elevation information (linework pixels). However for a good enough result a manual editing is needed (Figure 1-e).

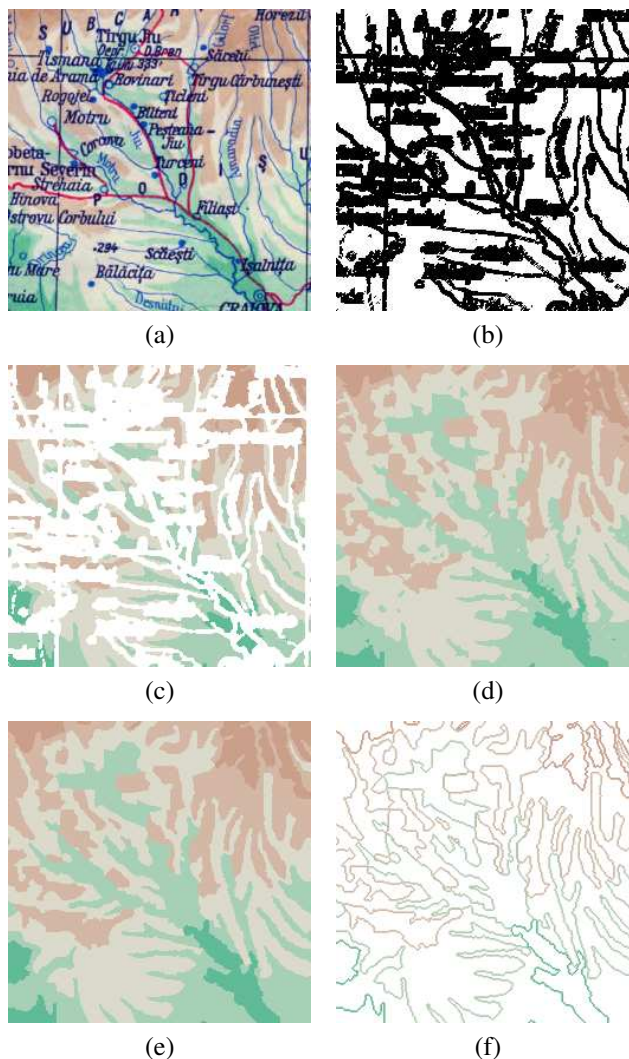


Fig. 1. Experimental results. (a) The preprocessed map image using the WHMM denoising algorithm; (b) The linework mask; (c) The color-clustered map using a vector median filtering and the minimum variance quantization algorithm; (d) Filled white gaps - a method based on a multi-pass majority filter algorithm; (e) The resulting image from Figure 1(d) – manually corrected; (f) The color-coded elevation contours.

6. ELEVATION CONTOURS EXTRACTION

Having a solid color-coded relief map with all unwanted information removed we can easily extract the elevation con-

tours by performing a sliding-neighborhood operation (Figure 1-f). Considering a 3 by 3 processing window, if at least one horizontal or vertical neighbor is different than central pixel, the color for the central element is preserved, otherwise the new color is white (not a border pixel). The color information of the contours is important for the future conversion into sparse elevation data based on the map's legend.

7. CONCLUSIONS

In this paper, we presented a method to extract the elevation contours from a color-coded relief map. The map was pre-processed using a WHMM based algorithm for denoising. Then a vector angle edge detector, with a saturation-based combination of hue and intensity planes, was applied. Each distinct elevation region was distinctly solid colored using median filtering and color clustering based on minimum variance quantization. A multi-pass majority filter algorithm was used to fill in the linework gaps. The color-coded contours were extracted by detecting the borders between colors through a sliding-neighborhood operation and they can be the input for an elevation data converter by means of the map's legend.

The method is computational expensive and cannot fully automate the process. But the approach can be a good help for the map digitizing domain when used with manual editing. Our future work may be concerned with the improving of the automation problem. Our final goal is to obtain a DEM (digital elevation model) and a 3D representation, using an interpolation method for filling in the space between the elevation contours.

8. REFERENCES

- [1] Patrice Arrighi and Pierre Soille, "From scanned topographic maps to digital elevation models," in *Proceedings of Geovision*, Belgium, 1999, International Symposium on Imaging Applications in Geology.
- [2] John Childs, "Development of a two-level iterative computational method for solution of the franklin approximation algorithm for the interpolation of large contour line data sets," M.S. thesis, Rensselaer Polytechnic Institute, Troy, NY, 2003.
- [3] Stephen Wise, "Scanning thematic maps for input to geographic information systems," *Computers & Geosciences*, , no. 21(1), pp. 7–29, 1995.
- [4] Hyeokho Choi and Richard G. Baraniuk, "Multi-scale image segmentation using wavelet-domain hidden markov models," *IEEE Transactions on Image Processing*, vol. 10, pp. 1309–1321, September 2001.

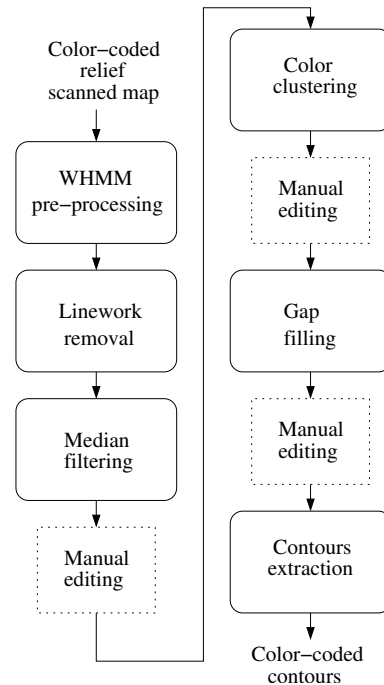


Fig. 2. The process of extracting the elevation contours from a color-coded relief scanned map.

- [5] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden markov models," *IEEE Transactions on Signal Processing*, vol. 46, pp. 886–902, April 1998.
- [6] Wesolkowski and E. Jernigan, "Color edge detection in rgb using jointly euclidean distance and vector angle," Trois-Riviere, Canada, May 1999, Vision Interface.
- [7] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proc. IEEE*, vol. 78, no. 4, pp. 678–689, April 1990.
- [8] Xiaolin Wu, "Efficient statistical computation for optimal color quantization," in *Graphics Gems II*, James Arvo, Ed. Academic Press, Boston, 1991.
- [9] Anil K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall International, Inc., 1989.
- [10] David Blatner, Glenn Fleishman, and Steve Roth, *Real world scanning and halftones*, Peachpit Press, Berkley, California, 2nd edition, 1998.