Extraction and Removal of Layers from Map Imagery Data

Alexey Podlasov and Eugene Ageenko

Department of Computer Science,
University of Joensuu,
P.O.Box 111, 80101 Joensuu, Finland
{apodla,ageenko}@cs.joensuu.fi

Abstract. Map images are composed of semantic layers depicted in arbitrary color. Layer extraction and removal is often needed for improving readability as well as for further processing. When image is separated into the set of layers with respect to the colors, it results in appearance of severe artifacts because of the layer overlapping. In this way the extracted layers differ from the semantic data, which affects further map image processing analysis tasks. In this work, we introduce techniques for extraction and removal of the semantic layers from the map images. The techniques utilize low-complexity morphological image restoration algorithms. The restoration provides good quality of the reconstructed layers, and alleviates the affect of artifacts on the precision of image analysis tasks.

1 Introduction

Nowadays, there exist various services delivering map imagery content on mobile devices. For example, map imaging applications provide user with a view of geographical map for the requested location. It could be also weather, traffic, pollution or any other kind of map. The imagery data is usually obtained from Digital Spatial Libraries [1], and transmitted via wireless network to user's computer or mobile device such as pocket PC, PDA, mobile phone, or similar mobile terminals. Map images need typically only a few color tones but high spatial resolution for representing details such as roads, infrastructure and names of the places. Though maps could be stored in vector format, raster map image is more preferable on a client-side since it is easier to transmit and handle. Raster images are also often used for digital publishing on CD-ROM and in the web.

The map images consist of a set of semantic layers, each containing data with distinct semantic content such as roads, elevation lines, state boundaries, water areas, temperature distribution, wind directions, *etc*. Layers are combined and displayed to the user as a generated color image, which is usually produced as follows. First the layers with different semantic nature are combined together by overlapping each other in a predefined order. Then the layers are depicted on the map with appropriate color and finally are transmitted to a client in an image form. After image has been received, client has no knowledge about the initial layer structure.

H. Kalviainen et al. (Eds.): SCIA 2005, LNCS 3540, pp. 1107 – 1116, 2005.

For example, we consider the topographic images from the NLS topographic database, in particular basic map series 1:20,000 [2]. These images consist of the following semantic layers: *Basic* (roads, contours, labels and other topographic data), *Elevation lines* (thin lines representing elevations levels), *Waters* (solid regions and poly-lines representing water areas and ways), *Fields* (solid polygonal regions), see Figure 1.

Though raster image is well suited for user observation, it cannot be easily used for further processing especially when semantic data is required. For example, when one needs to calculate the area of fields or e.g. the length of sea shore the semantic layer corresponding to the water or field areas must be obtained first. The layers can be extracted from the raster map image through *color separation* process. During this process, the map image is divided into binary layers each representing one color in the original image. The problem is that the separation introduces severe artifacts in places where one layer overlaps another, see Figure 2. These artifacts make separated layer inappropriate for many image analysis tasks. In order to use corrupted layers in further processing a restoration technique should be designed.

Another task is to remove some irrelevant layer(s) from the map image. For example, a car driving user does not need elevation lines on a map. Their presence impairs map readability, which can be improved by the layer removal. Since elevation lines are drawn on the fields and waters, one must apply reconstruction to remove artifacts left by the removed layer.

Moreover, it has been shown that the best compression results for raster map image can be achieved if the image is decomposed into binary semantic layers, which are consequently compressed by the algorithm designed to handle binary data (e.g. JBIG) [3]. Color separation artifacts affect the statistical properties and consistency of the layers, and result in degraded compression performance.

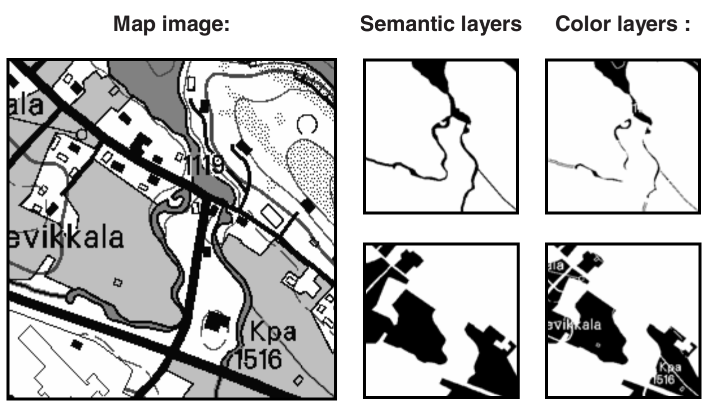

Fig. 1. Illustration of map image, its semantic structure, and color layers showing the artifacts due to color separation (with permission of National Land Survey of Finland)

The approximation of the noise-corrupted image to the original is often achieved by using various noise removal techniques, image enhancement or statistical analysis

[4-10], These approaches are limited to a local neighborhood only, and cannot exploit non-local semantic properties of the image. Semantic approaches exploiting global properties of the image typically have high complexity and are suitable for very special kind of data e.g. thin line graphics or text [11-13]. Existing noise filtering and image enhancement algorithms could not be considered for our problem.

In this work we present two techniques for extraction and, correspondingly, removal of the semantic layers from raster map images using color separation process. The algorithms are based on the morphological restoration algorithms that attempt to recover the semantic layer structure from the separated color layer. The technique is applied for analysis of the semantic data as well as for (on demand) removal of irrelevant semantic content from the map image. The effect of the restoration is limited only to the areas which are degraded due to separation and would be overlapped with other layers during composition. Therefore the color image obtained using combination of the restored layers matches exactly the initial image without any degradation in the quality. Due to simplicity of morphological operations, the method is also fast and simple to implement on the modern mobile devices.

The rest of the paper is organized as follows. Mathematical morphology is briefly introduced in Section 2. Then in Section 3, we introduce new filtering method for layer extraction, and then apply it for layer removal in Section 4. Empirical results are reported in Section 5, and conclusions drawn in Section 6.

2 Mathematical Morphology Fundamentals

Mathematical morphology [13] refers to a branch of nonlinear image processing and analysis originally introduced by Georges Matheron [14] and Jean Serra [15]. In mathematical morphology, the binary image space E is defined as $E = \mathbb{Z}^2$ (the space of all possible image pixel locations), and the binary image X as a set $X \subseteq E$. The main principle of mathematical morphology is to analyze geometrical and topological structure of an image X by “probing” the image with another small set $A \subseteq E$ called a structuring element. The choice of the appropriate structuring element depends on the particular application.

Let us define the dilation of X by A, denoted by $\delta A(X)$, as an operator on $\mathcal{P}(E)$ such as:

$$\delta_A(X) = \bigcup_{a \in A} X_a = \{h \in E \mid \widetilde{A_h} \cap X \neq \varnothing\}, \tag{1}$$

The *erosion of X by A*, denoted by $\varepsilon_A(X)$, is consequently:

$$\varepsilon_A(X) = \bigcap_{a \in A} X_{-a} = \{h \in E \mid A_h \subseteq X\}, \tag{2}$$

where $\tilde{A} = -A = \{-a \mid a \in A\}$ is the reflectance of A with respect to the origin. Let us also define the translation invariant operator $\rho_{A,n}$ called *rank operator* as follows:

$$\rho_{A,n}(X) = \left\{h \in E \,\middle|\, card\left(X \cap A_h\right) \geq n\right\}. \tag{3}$$

The operator $\rho_{A,n}(X)$ sets current pixel to be foreground if the amount of foreground pixels in a neighborhood defined by the structuring element is greater than n. Otherwise the pixel is defined as a background pixel. Since rank operator performs similar to erosion or dilation depending on the value of the rank parameter, it is possible to treat the rank as soft counterpart of classical erosion and dilation operators. In particular

$$\delta_{\tilde{A}}(X) = \rho_{A,1}\left(X\right) \text{ and } \varepsilon_A(X) = \rho_{A,n}\left(X\right). \tag{4}$$

Sometimes it is important to restrict the area where operator could be applied. This can be accomplished by using conditional operators: if image A is a subset of image M, then for any operator $\psi(A)$ the operator $\psi(A \mid T)$ is called $\psi(A)$ conditional relative to mask image M and is defined as follows:

$$\psi(A \mid T) = \psi(A) \cap T\ . \tag{5}$$

3 Layer Extraction

When original semantic data is unavailable, the task of restoration leaves a lot of freedom for algorithm designer as one can only guess the initial layer structure. The only restriction we have is that the composition of reconstructed layers would be identical to the initial color map. In other words, we can modify the value of the pixels in the layers only if the same pixel value is set in one of the higher priority (overlapping) layers. This means that the change of the pixel value will be seen only in the particular layer, but not in the color image corresponding to the reconstructed layers.

3.1 Algorithm Structure

The algorithm consists of three principal steps: decomposition, mask creation and layer restoration, as outlined in Figure 2. At the first step, the color map image (scanned or obtained from the third party source) is decomposed into a set of binary layers by color separation process. This is done so that each layer represents one color in the original image [3]. On the second step, we define a mask – an area where reconstruction could be performed restricting the reconstruction of the layers to be equal to the original color image. Finally, the layer is extracted and restored using the proposed restoration algorithm. Further we describe in details second and third steps of the algorithm.

3.2 Mask Construction

The conditioning mask defines the set of pixels that are allowed to change in the restoration so that the combination of the restored layers would be kept untouched. Since we have assumed that the order of layer overlapping is predefined, the mask for every layer will be a union of all upper-laying layers, see Figure 3. All modifications made to the pixels within the mask area will be overlapped when the combined color image is represented to the user. Depending on the particular case, it is possible to simplify the mask structure by taking into account the nature of the objects represented on the map. For example, we can expect that Waters and Field layers cannot overlap in reality, and therefore, could not overlap on a combined map image. When implementing, we can exclude these layers from the conditioning mask (see Figure 4).

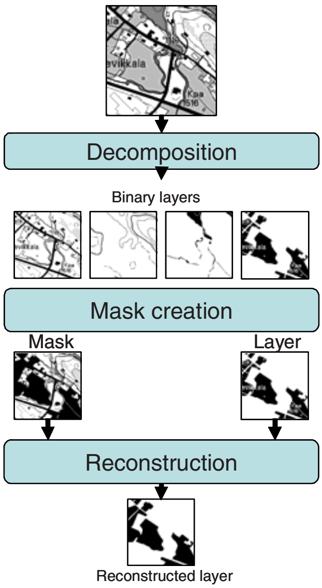

Fig. 2. The diagram of the layer extraction algorithm

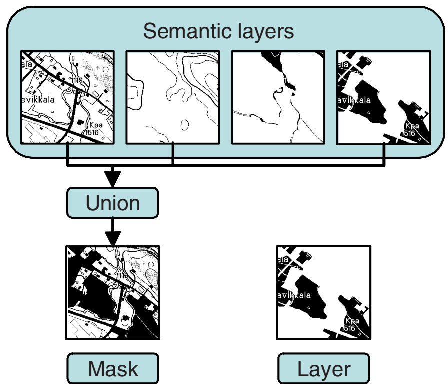

Fig. 3. The approach for the mask construction

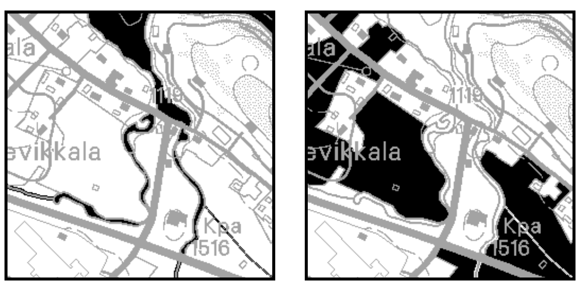

Fig. 4. Water and Fields layers with their masks. Object pixels are shown in black, mask pixels in gray color, and background in white

3.3 Layer Restoration

We reconstruct layer iteratively. With every iteration, the object areas spread within the mask, and then the mask area shrinks. The spreading is performed by dilation operator δA(X) and mask shrinking by erosion operator εA(X). The pseudo-code of the layer restoration algorithm is shown in Figure 5, and its diagram is outlined in Figure 6. The stepwise process of the iterations is illustrated in Figure 7.

REPEAT

$L := \delta_A(L \mid M)$;

$M := \varepsilon_B(M)$;

$M := L \cup M$;

UNTIL *Iteration criterion met*

Fig. 5. Outlined of the layer restoration algorithm

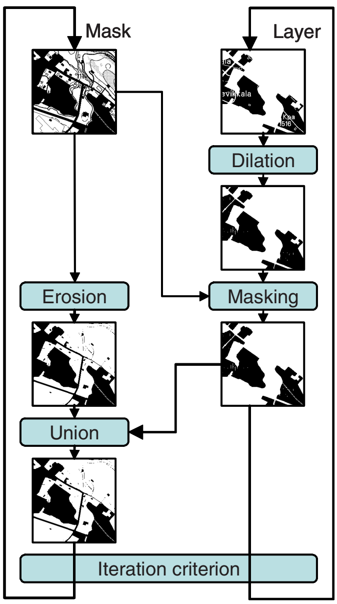

Fig. 6. Block diagram of the layer restoration algorithm

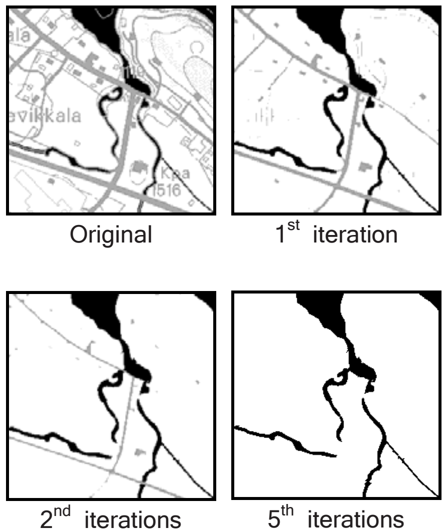

Fig. 7. Step-by-step illustration of the dilation with mask erosion. Pixels of the processed object are marked in black, whereas the pixels belonging to the mask – in gray

The iterative process is controlled by a stopping criterion. We have investigated two approaches: *Iterate until stability* and *Iterate fixed amount of times*. The first approach assumes that the iterative process will continue until the layer (and mask) converges. The convergence is guaranteed because the erosion sequentially decreases the mask, see Figure 7. We can therefore perform the iterations until the mask equals to the layer itself.

Examination if the mask and layer are equal could be a time consuming operation, especially if the image size is big. To avoid this, we consider the second approach assuming that most of the artifacts being of limited size. Therefore it is sufficient to perform a predefined (small) number of iterations to complete the restoration process. For example, if we suppose that the size of an artifact is less than 4 pixels, on average, only 3 iterations with 3×3 block are needed.

As with the conditional closing, an important question is the choice of an appropriate structuring element. There are two structuring elements used in the algorithm. By varying the element used for dilation we can control how fast the object expands over the mask, while varying the element used for erosion we control how fast the mask shrinks. An essential matter is the relationship between the dilation and erosion speeds. Let A be the structuring element of dilation and B be the structuring element of erosion. In our investigations, we have tested three cases: objects dilating faster than mask eroding ($A = block_{3\times3}$, $B = cross_{3\times3}$), objects dilating slower than mask eroding: ($A = cross_{3\times3}$, $B = block_{3\times3}$), and the case of equal speed ($A = block_{3\times3}$, $B = block_{3\times3}$ or $A = cross_{3\times3}$, $B = cross_{3\times3}$).

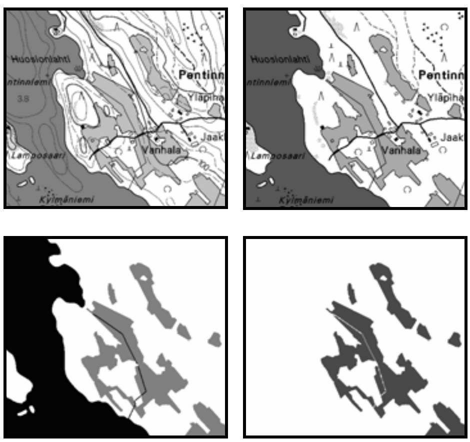

Fig. 8. Example of the consecutive layer removal (map image fragment, elevation lines removed, basic layer removed, water areas removed)

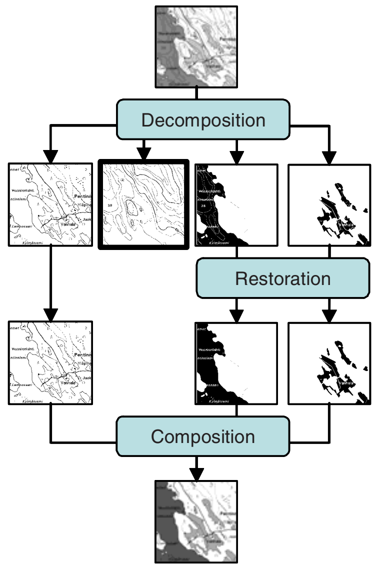

Fig. 9. Block diagram of the layer removal algorithm. Elevation lines layer to be removed is outlined with a black frame

4 Layer Removal

The task of layer removal arises when less important layers are needless to the map user, e.g. user driving a car does not need elevation lines. In order to remove a layer, the restoration technique described in Section 3 is first applied to all underlying layers in order of overlapping. Then the restored layers except the removed one are composed into the color image, see Figure 9. The most important criterion here is the quality of the restoration – how closely the restored layer approximates the semantic data. Moreover, in interactive applications the visual appearance of the reconstructed layer becomes essential. Figure 8 illustrates the effect of the successive removal of *Elevation*, *Basic* and *Water* layers.

5 Evaluation

The restoration technique has been evaluated on a set of topographic color-palette map images. These images were decomposed into binary layers with distinctive semantic meaning identified by the pixel color on the map. The restoration algorithm has been applied for reconstruction of these semantic layers after the map decomposition process. Both the combined color map images and the binary semantic layers composing these color map images were originally available for testing. This gave us a possibility to compare restored images with their original undistorted counterparts.

The test set consists of five randomly chosen images from the "*NLS Basic Map Series 1:20000*" corresponding to the map sheets No. 431306, 201401, 263112, and 431204. Each image has dimension 5000×5000 pixels and corresponds to 10×10 km area. Images are composed of four semantic layers:

- *Basic* –buildings, protected sites, benchmarks and administrative boundaries;
- *Elevation* – elevation lines;
- *Water* – lakes, rivers, swamps, water streams;
- *Fields* – agricultural areas.

In the following we evaluate the proposed technique by estimating the restoration quality using image *similarity measurement*, *area measurement* and *length of the sea shore*. For image similarity we consider *Normalized Mean Absolute Error* (NMAE), which is a Hamming Distance between images computing the average number of different pixel values. In our context, we compare the original (not affected by decomposition) semantic layers and the layers restored using the proposed technique:

$$NMAE(X,Y) = \frac{\sum_{j=1}^{\mathbb{H}} \sum_{i=1}^{\mathbb{W}} \left| x_{i,j} - y_{i,j} \right|}{\mathbb{H} \cdot \mathbb{W}}, \tag{6}$$

where $\mathbb{H}$ and $\mathbb{W}$ are image dimensions.

We measure NMAE difference between reconstructed *Waters* and *Fields* layers and the original ones and show the improvement comparing to the corrupted layers. We present obtained NMAE difference for every layer separately in total within the test set (see Figure 10).

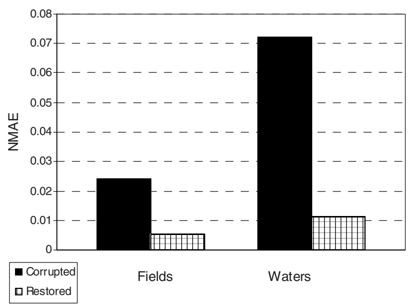

Fig. 10. The average NMAE difference with the original measured for restored. Fields (left) and Waters (right) layers comparing to corrupted ones

In the following, we compare area measured over the original layer with one measured over reconstructed and corrupted layer. The results are presented for Waters and Fields layers separately on average within the test set, see Table 1. Reconstruction reduces the error of the area measurement from near 15-20% to just about 1%. The length of the sea shore is measured as the length of object borders in Waters layer. Results – the length and error over original, corrupted and reconstructed layer are represented in Table 2. Reconstruction reduces the error of shore length calculation from 37% to 1%.

Table 1. The area (in pixels) and error comparing to original value (in percents) measured over original, corrupted and reconstructed Waters and Fields layers

Layer	Semantic layers	Corrupted layers		Reconstructed layers	
	Area	Area	%	Area	%
Waters	10 480 893	8 678 605	17.2	10 389 501	0.8
Fields	4 267 983	3 663 960	14.1	4 262 378	0.1

Table 2. The length of the sea shore and error comparing to original value (in percents) measured over original, corrupted and reconstructed Waters layer

Semantic layers	Corrupted layers		Reconstructed layers	
Length	Length	%	Length	%
3 115 505	4 279 979	37.3	3 074 954	1.3

6 Conclusions

A technique for the extraction and removal of semantic layers from map imagery data has been proposed. The extracted semantic data can be further used for various image analyzing and processing tasks (e.g. area measurement); whereas the layer removal is useful for removing unwanted data from map images due to various reasons (e.g. view cluttering). The proposed technique is based on the separation of the raster map image into color layers and subsequent elimination of the artifacts caused by the color separation process. The iterative restoration algorithm based on the conditional morphological operators is designed for layer reconstruction. The performance of the proposed technique is evaluated qualitatively by comparing the reconstructed layers with the native semantic data, and quantitatively by using standard image analysis tasks. Quality evaluation demonstrates that restoration algorithm can efficiently approximate the map layers. When properly tuned, the algorithm reduces the error in such image analyzing applications as area measurement from 15-20% to about 1%. The reconstructed layers have lesser entropy and can substitute for the color layers in map data storage without any loss of quality. It is possible because the restoration is limited to the area of the images that are overlapped by other layers. Therefore the color raster map image can be obtained by the combination of the reconstructed layers and still remain absolutely identical to the initial non-processed map image.

References

1. Fox E.A., et al. (Eds.) “Digital Libraries”. [Special issue of] *Communications of the ACM* 38 (4), 1995.
2. NLS: National Land Survey of Finland, Opastinsilta 12 C, P.O.Box 84, 00521 Helsinki, Finland. http://www.nls.fi/index_e.html.
3. Fränti P., Ageenko E., Kopylov P., Gröhn S. and Berger F., "Compression of map images for real-time applications", *Image and Vision Computing*, 22 (13), 1105-1115, November 2004.
4. Pitas, I., Venetsanopoulos A.N., *Nonlinear digital filters: principles and applications*, Boston, Mass.: Kluwer, 1990.
5. Dougherty E.R., Astola J. (eds) *Nonlinear Filters for Image Processing*, SPIE Optical Engineering Press, 1997.
6. Dougherty E.R., “Optimal mean-square n-observation digital morphological filters. Part I: Optimal binary filters”, *Computer Vision, Graphics, and Image Processing*, 55: 36-54, 1992.
7. Wah, F.M., “A binary image preprocessor for document quality improvement and data reduction”, *Proc. Int. Conf. on Acoustic, Speech, and Signal Processing-ICASSP’86*, 2459-2462, 1986.
8. Ping Z., Lihui C., Alex K.C., “Text document filters using morphological and geometrical features of characters”, *Proc. Int. Conf on Signal Processing-ICSP’00*, pp. 472-475, 2000.
9. Randolph T.R., Smith M.J.T., “Enhancement of fax documents using a binary angular representation”, *Proc. Int. Symp. on Intelligent Multimedia, Video and Signal Processing*, pp. 125-128, Hong Kong China, 2-4 May 2001.
10. Zheng Q., Kanungo T., “Morphological degradation models and their use in document image restoration”, University of Maryland, USA, Technical Report, LAMP-TR-065 CAR-TR-962 N660010028910/IIS9987944, 2001.
11. Ageenko E., Fränti P., “Context-based filtering of document images”, *Pattern Recognition Letters*, 21 (6-7), 483-491, Elsevier Science, 2000.
12. Kolesnikov A., Fränti P., Data reduction of large vector graphics, *Pattern Recognition*, 38(3), 2005, pp 381-394.
13. Heijmans H.J.A.M., *Morphological image operators*. Boston: Academic Press, 1994.
14. Matheron G. *Random Sets and Integral Geometry*, J. Wiley & Sons, New York, 1975.
15. Serra J., Image Analysis and Mathematical morphology. London: Academic Press, 1982.