

# ON THE RECONSTRUCTION OF SEMANTIC DATA IN RASTER MAP IMAGES

Eugene Ageenko and Alexey Podlasov

*Department of Computer Science, University of Joensuu, Box 111, FIN-80101 Joensuu, Finland  
ageenko@cs.joensuu.fi, apodla@cs.joensuu.fi*

## ABSTRACT

Raster map images consist of a set of layers depicted in arbitrary color. There exist strong correspondence between the color of the layer and its semantic meaning. Often there is a need to separate or extract semantic layers from the maps. The separation results in severe artifacts in places where semantic layers would overlap (e.g. elevations lines drawn on top of the topographic map). In the current work, we design the technique to reconstruct the semantic layers from the color layers resulting from the image separation process. The proposed technique provides good visual quality of the reconstructed image layers, and can therefore be applied for selective layer removal/extraction, which is often necessary in map processing and analyzing applications. It improves the accuracy of the data analysis and measurement tasks. It also alleviates compression deficiency of reconstructed layers versus corrupted ones with lossless compression algorithms (ITU Group 4, PNG, JBIG). The technique requires few computation resources and can therefore be successfully used in mobile computers and terminals.

## KEYWORDS

Map, images, semantic, restoration, morphology, compression.

## 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.

Raster map images upon creation were composed of a set of layers, each containing data with distinct semantic content such as roads, elevation lines, state boundaries, water areas, temperature distribution, wind directions, *etc.* Layers were combined and displayed to the user as a generated (color) raster image. For example, we consider the topographic images from the NLS topographic database, in particular basic map series 1:20,000 [2]. The basic set contains 4 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 vector format is more convenient for storing and processing of semantic layers forming a database, it is significantly more resource demanding. Generated 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 or in the web. This raises the problem of huge storage size of the map images. Especially it is apparent in applications requiring the use of mobile hardware such as mobile phones or pocket computers. For example, 10 seconds transmissions via GPRS channel with maximal possible bandwidth 45kb/sec results in at most 54kB of image data. This corresponds to only about 500×500 pixels 4-layer map image. In practice transmission speeds are about ten times slower.

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, the semantic layer corresponding to the field areas must be first obtained. 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 semantic layer has overlapped another during map composition, see Figure 2. These artifacts make separated layers inappropriate for many image analysis tasks. In order to use corrupted layers in further processing a restoration technique should be designed. 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 [3]. Color separation artifacts affect the statistical properties and consistency of the layers, and result in degraded compression performance. Especially it is apparent in applications requiring storage on mobile hardware such as mobile phones or pocket computers. Larger image size also takes longer and expensive to transmit.

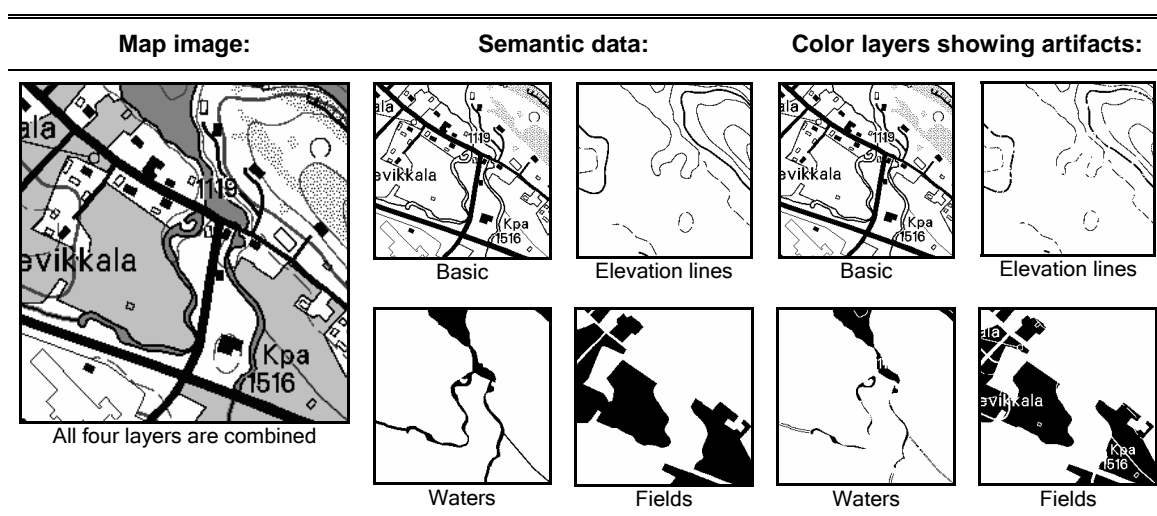


Figure 1. Illustration of map image, its semantic structure, and color layers showing the artifacts due to color separation (with permission of National Land Survey of Finland).

In the early development of signal and image processing, linear filters were the primary tools. However, linear filters have poor performance in the presence of noise as well as the problems where system nonlinearities are encountered [4-5]. In the early sixties investigations of Matheron and Serra led to a new quantitative approach in image analysis, nowadays known as mathematical morphology [6-7]. One of the primary applications of mathematical morphology is noise removal. It was established in the classical works on mathematical morphology such as [6], and continues to develop nowadays [8,9]. Also, it has been found that morphological processing of the image could be used to decrease the entropy of the image, and therefore increase the compression performance. The idea of using nonlinear filters for improving compression performance of the image is known as image enhancement and was investigated in [10-14]. However, mathematical morphology is not a universal tool because morphological operators manage within the local window and can not take into account global properties of the image. Several methods have been considered for image processing by analyzing the local pixel neighborhood defined by a filtering template. Techniques have been proposed based on the analysis of the context information [15-17]. Most of previously developed methods are not directly applicable to the context of map images, others (as in the case of latter techniques) require significant computational or memory resources which are not available for mobile devices. Also we must take into account that most filtering techniques improve compression performance by degrading the image. However, in our task we can not use that kind of enhancement because we agreed that the content of the map image is critically important and can not be degraded.

Therefore we have developed specific lossless reconstruction technique dedicated for the map images. In order to utilize this technique on mobile hardware we have restricted ourselves with computational complexity and choose mathematical morphology as a simple and effective filtering and enhancement tool. The effect of the reconstruction 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 reconstructed layers matches exactly the initial image without any degradation in the quality.

## 2. MATHEMATICAL BACKGROUND

### 2.1 Mathematical morphology

*Mathematical morphology* refers to a branch of nonlinear image processing and analysis originally introduced by Georges Matheron [7] and Jean Serra [6]. In mathematical morphology, the binary image space  $E$  is defined as  $E = Z^2$  (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  $\Pi(E)$  such as:

$$\delta_A(X) = \bigcup_{a \in A} X_a = \{h \in E \mid \exists \bar{h} \cap X \neq \emptyset\}, \quad (1)$$

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

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

where  $A_0 = -A = \{-a \mid a \in A\}$  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:

$$\rho_{A,n}(X) = \{h \in E \mid \text{card}(X \cap A_h) \geq n\}. \quad (3)$$

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. From Matheron’s representation theorem [7] it follows that rank operator can be treated as the base operator of mathematical morphology. 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

$$\delta_{A_0}(X) = \rho_{A,1}(X) \text{ and } \varepsilon_A(X) = \rho_{A,n}(X), \quad (4)$$

The operator  $\alpha_A(X) = \delta_A(\varepsilon_A(X))$  is called the *(structural) opening* by  $A$ .

Dually, the operator  $\beta_A(X) = \varepsilon_A(\delta_A(X))$ , is called the *(structural) closing* by  $A$ .

### 2.2 Conditional operators

If an image is, say, dilated by a structuring element containing the origin, it is expanded, and the manner of the expansion depends only on the shape of the structuring element. If the dilation is successively repeated, the original image grows without bound. Sometimes it is important to restrict the growth. This can be accomplished by using conditional operators. A common form of conditioning restricts the translations to a superset of the input image: if image  $A$  is a subset of image  $T$ , then for any operator  $\psi(A)$  the operator  $\psi(A|T)$  is called  $\psi(A)$  *conditioned relative to  $T$*  and defined as:

$$\psi(A|T) = \psi(A) \cap T. \quad (5)$$

The image  $T$  is usually referred to as *mask* image.

### 3. SEMANTIC LAYER RECONSTRUCTION

#### 3.1 Formal problem definition

When original semantic data is unavailable, the task of reconstruction leaves a lot of freedom for algorithm designer since one can only guess about semantic layer structure. When we got a (color) raster image, we need to decompose it first to layers using color separation process as described in Introduction. Let us define *decomposition* as the process  $M \xrightarrow{D} M$  of splitting N-color map image  $M$  into a layer set  $M = \{L_1 \dots L_N\}$  by separating pixels to different layers by their color value. It can be easily shown that layer set once composed into raster map image could not be reconstructed back:

$$\forall M_1 = \{L_1, \dots, L_N\} \xrightarrow{C} M \xrightarrow{D} M_2 = \{L_1^*, \dots, L_N^*\} \subseteq M_1, \tag{6}$$

where  $M \xrightarrow{C} M$  is the process of composition of layers back into a map image

The task of reconstructing the layer  $L_k$  is to construct an operator  $\psi(L_k)$  such that

$$\begin{cases} \psi(L_k) : P(E) \rightarrow P(E) \\ \{L_1^*, \dots, L_k^*, \dots, L_N^*\} \xrightarrow{C} M_1 \\ \{L_1^*, \dots, \psi(L_k), \dots, L_N^*\} \xrightarrow{C} M_2 = M_1 \end{cases} ; \tag{7}$$

The choice of operator mainly depends on the application area. The main criteria will be to minimize the difference between  $\psi(L_k)$  and  $L_k$ . The only restriction we set is that the composition of reconstructed layers would be identical to the initial color map. The aim of condition  $M_2 = M_1$  is therefore to protect the original raster map image from any degradation during the restoration. This requirements can be obeyed conditioning an operator  $\psi(L_k)$  on the mask  $T_k$ , which defines an area where changes of the layer content are allowed. Keep in mind that composition of binary semantic layers must be preserved unchanged. Therefore restoration operator cannot remove pixels, which are already present on the corrupted layer. It can only add pixels to a layer, so that the condition (8) is maintained:

$$L_k \subset \psi(L_k | T_k). \tag{8}$$

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 2. 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 3.

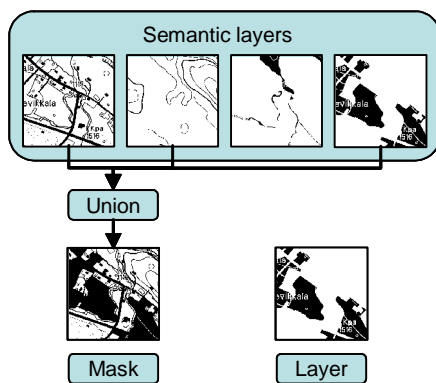


Figure 2. The approach for the mask construction.

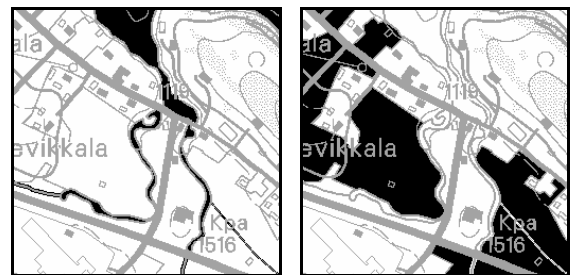


Figure 3. Water and Fields layers with their masks. Object pixels are shown in black, mask pixels in gray color, and background in white

### 3.2 Layer reconstruction

We define a (gray-scale or color) map image as a pair

$$M = \left( \begin{array}{l} C \subseteq E \\ c: \mathbf{Z}^2 \rightarrow \mathbb{Y} \end{array} \right), \quad (9)$$

where  $C$  is a set of pixel locations (a binary image) and  $c(h)$  is intensity (or color) function which yield a gray value or an index into a color palette. Our reconstruction algorithm (referred further as *CDME*) takes this image as an input and generates a set of reconstructed semantic layers  $\hat{M} = \{\hat{L}_1 \dots \hat{L}_N\}$  as output.

The pseudo-code of the algorithm is shown in Figure 4. First the map image is decomposed into a set of the color layers  $M = \{L_1 \dots L_N\}$ . Then for each layer the mask is constructed, and the layer restoration operator  $\hat{L}_k = \psi(L_k | T_k)$  is applied iteratively. At every iteration, the object areas spread within the mask, and then the mask area shrinks. The spreading is performed by dilation operator  $\delta_A$  and mask shrinking by erosion operator  $\varepsilon_A$  respectively. The diagram of restoration operator is outlined in Figure 5, and the stepwise process of the iterations is illustrated in Figure 6. 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 6. 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 have 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  $3 \times 3$  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: in the objects dilation and in the mask erosion. With varying the first element we can control how fast the object expands over the mask, while varying the second element controls how fast the mask shrinks. An essential matter is the relation between the speeds of the dilation and erosion. 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 ( $A = \text{block } 3 \times 3, B = \text{cross } 3 \times 3$ ), objects dilating slower than mask eroding: ( $A = \text{cross } 3 \times 3, B = \text{block } 3 \times 3$ ), and the case of equal speed ( $A = \text{block } 3 \times 3, B = \text{block } 3 \times 3$  or  $A = \text{cross } 3 \times 3, B = \text{cross } 3 \times 3$ ).

The reconstructed semantic layers can be combined back into the color map image  $\hat{M}$  as follows:

$$\hat{M} = \left( \begin{array}{l} C = \bigcup_{k=1}^N \hat{L}_k \\ c(h) = \min_{\forall k | h \in \hat{L}_k} k \end{array} \right), \quad (10)$$

It can be easily proven that the resulting combined map image is identical to the source map image:

$$M \xrightarrow{D} \mathbf{M} \xrightarrow{\psi} \hat{M} \xrightarrow{C} \hat{M} = M, \quad (11)$$

Input: $N$ -color map image $M$	
<pre> FOR <math>k \leftarrow 1</math> TO <math>N</math> DO   <math>L_k \leftarrow \{h \in C \mid c(h) = k\}</math>    <math>T_k \leftarrow \bigcup_{j=1}^k L_j^*</math> END FOR </pre>	<p>Preparation stage:</p> <p>layer decomposition <math>M \xrightarrow{D} \mathbf{M} = \{L_1 \dots L_N\}</math></p> <p>mask <math>T</math> creation</p>

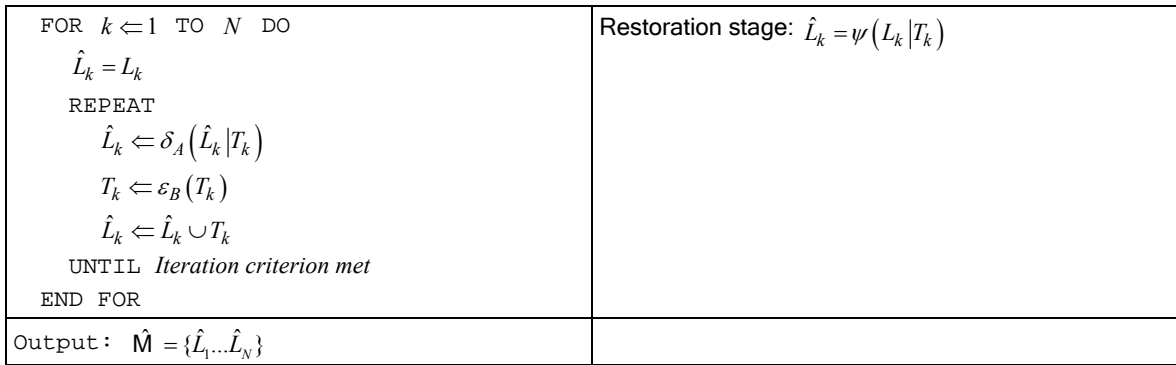


Figure 4. Layer reconstruction algorithm, here  $\leftarrow$  is the assignment operator

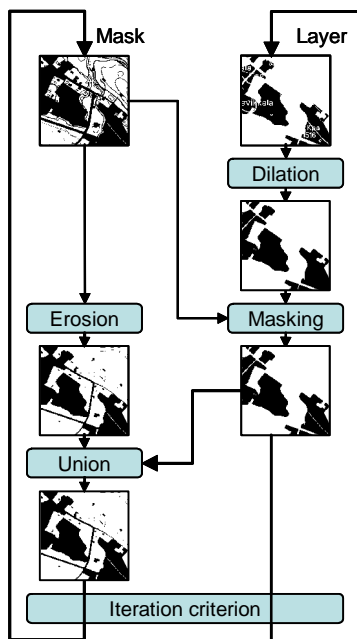


Figure 5. Block diagram of the layer restoration algorithm.

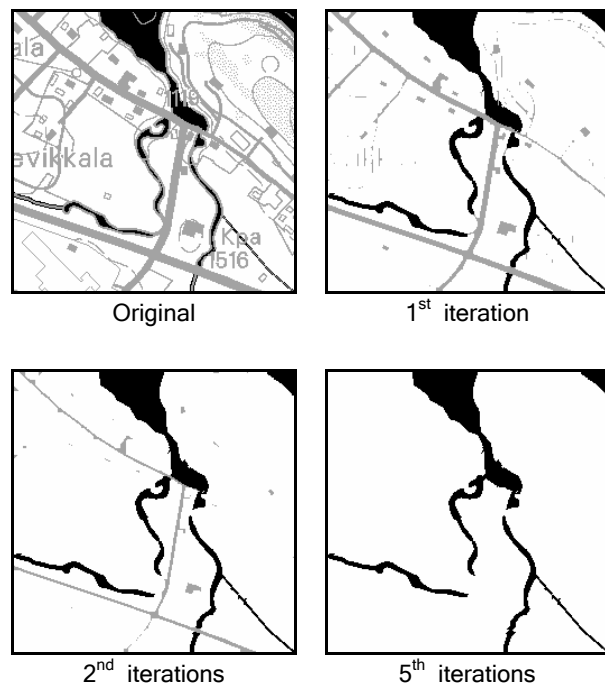


Figure 6. Step-by-step illustration of the dilation with mask erosion. Pixels of the processed object are marked in black, whereas the pixels belonging to the mask – in gray.

## 4. EVALUATION

In this section we consider three application areas and perform empirical evaluation of the reconstruction technique. As a test set we use topographic color-palette map images from the “*NLS Basic Map Series 1:20000*”. Images are composed of four semantic layers: Basic topographic data, Elevation lines, Water areas and Fields of agriculture. Both, the color raster maps and the semantic layers composing these color maps, were originally available for testing. This fact gave us possibility to compare restored images with their original undistorted counterparts. The structure of Elevation lines layer cannot be completely reconstructed using the proposed techniques, therefore we have not applied the reconstruction for this layer.

By restoration quality could be considered any particular measure of how close to the original corrupted layers were reconstructed. The simplest way is to measure the difference between the original (undistorted) semantic layers and the reconstructed semantic layer using the proposed technique. In particular, we present here results as *Normalized Mean Absolute Error* (NMAE), which is the Hamming Distance that measures the average number of different pixel values in two images relatively to the image size. To further demonstrate the usability of the method, we compare the results with the reconstruction based on non-iterative operator referred further as *conditional closing*  $\zeta$ , where  $C$  is a  $7 \times 7$  pixel block:

$$\zeta(L_k) \leftarrow \beta_C(L_k | T_k) \tag{12}$$

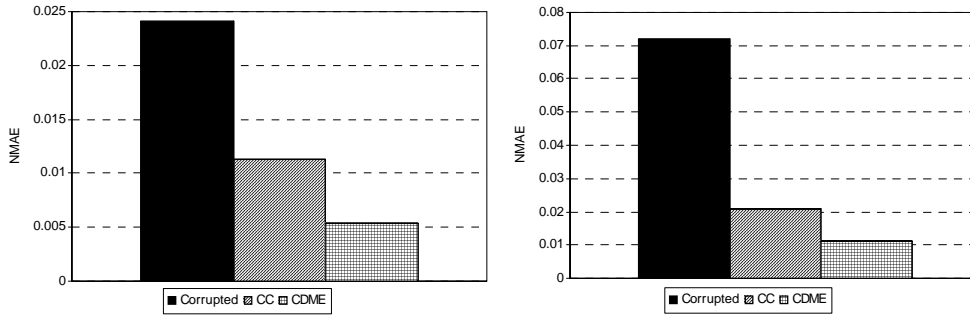


Figure 7. The difference between the semantic and color layers (Corrupted) and semantic and reconstructed semantic layers with  $\zeta$  (CC) and  $\psi$  (CDME) restoration operators for Fields layer (left image) and Water layer (right image).

### 4.1 Area measurement

In the following, we compare area measured over the original semantic layer with one measured over reconstructed semantic and corrupted (color) layer. The results are presented for Water and Fields layers separately on average within the test set, see Table 1. Reconstruction reduces the error of the area measurement to less than 1%. Since CDME approximates layers better, its area measurements are also much closer to the original than the CC results.

Table 1. The area (in square meters) measured over original semantic layers, color layers, and reconstructed semantic layers with CC and CDME algorithms for Water and Fields layers.

Image Layer	Semantic layers		Color layers		Reconstructed $\zeta$ (CC)		Reconstructed $\psi$ (CDME)	
	Area	Area	Area	Error, %	Area	Error, %	Area	Error, %
Water	41,923,572	34,714,420	39,985,816	17.2	41,558,004	4.62%	41,558,004	0.8
Fields	17,071,932	14,655,840	16,229,012	14.1	17,049,512	4.94%	17,049,512	0.1

### 4.2 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. Such layers can embarrass map readability and must be removed. In order to remove a layer, the restoration technique described first applied to all underlying layers in order of overlapping. Then the restored layers except the removed one are composed into the color image. 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. And the quantitative evaluation results are shown in Table 2, in which the quality of the map image with removed layers composed upon reconstructed layers is compared versus the map image composed upon the color layers. The quality is evaluated as the percentage of the mismatch pixels in comparison with the map composed upon semantic layers.

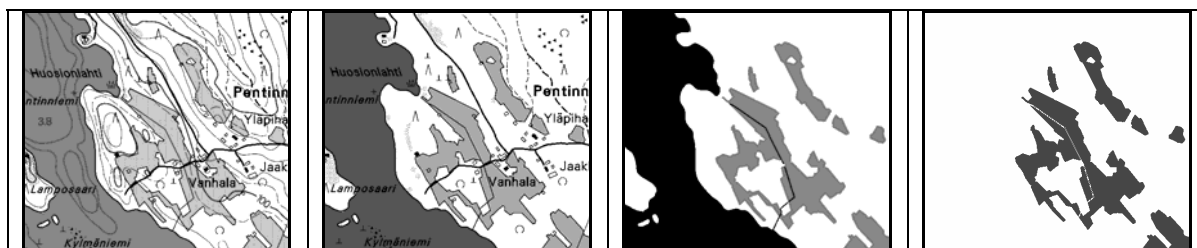


Figure 8. Example of the consecutive layer removal.

Left-to-right: original map image fragment, elevation lines removed, basic layer removed, water areas removed.

Table 2. Quality error computed for the map images with removed layers constructed upon Color and Reconstructed layers, in percents.

Error is computed for	Layers that have been removed		
	Elevation	Elevation + Basic	Elevation + Basic + Water
Color layers:	3.08	15.15	4.70
Reconstructed layers:	0.72	2.62	1.22

### 4.3 Layer compression

In this evaluation we examine the compression performance of the layers. The proposed technique is evaluated against three compression techniques: LZ (PNG), ITU Group 4 (TIFF) and JBIG. Tables 3 and 4 represent the average compressed sizes of restored Water and Fields layers correspondingly. The results given in the table are: the average compressed file size (*Size*) and improvement ratio (*Improvement*) for original semantic layers, corrupted color layers and reconstructed semantic layers. We conclude that proposed algorithms improve the compression performance of corrupted map layers. The measure of improvement for every single layer depends on the applied compression technique and varies approximately from 20 to 50%.

Table 3. Compression results (in bytes) for reconstructed Water layer

Compression algorithm	Semantic layer	Color layer	Reconstructed layer	
			Size	Improvement
PNG	381,608	425,862	384,766	9.65 %
TIFF	167,361	357,164	168,630	52.79 %
JBIG	81,334	137,258	93,230	32.08 %

Table 4: Compression results (in bytes) for reconstructed Fields layer

Compression algorithm	Semantic layer	Color layer	Reconstructed layer	
			Size	Improvement
PNG	309,712	456,710	320,821	29.75 %
TIFF	99,622	196,456	105,388	46.36 %
JBIG	49,409	113,977	50,936	55.31 %

## 5. CONCLUSIONS

A technique for the reconstruction of the semantic layers extracted from the raster map images has been proposed. The extracted semantic data can be further used for various image analysis (e.g. area measurement) and storage applications. The layer removal is useful for removing unwanted data from map images due to various reasons (e.g. view cluttering). The proposed technique uses iterative restoration algorithm based on

mathematical morphological filters specially designed for layer reconstruction. The performance of the proposed technique is assessed qualitatively and quantitatively by comparing the reconstructed layers with the native semantic data, and evaluation of the use of reconstructed layers in typical image analysis and compress applications. Quality evaluation demonstrates that restoration algorithm can efficiently approximate the map layers. The technique is able to reduce the error in such image analyzing applications as area measurement from 15-20 % to about less than 1 %, and accuracy of layer removal from over 15 % to less than 3 %. The reconstructed layers have lesser entropy and can substitute for the color layers in map data storage without any loss of quality and improve compression performance of reconstructed layers by 10-50 %. By design, the layer reconstruction is limited to the area of the images that are overlapped by other layers. Therefore the color raster map image obtained by the combination of the reconstructed layers will be identical to the source map image. Due to low complexity, proposed technique can be easily implemented on the computers with limited computational resources. For implementation details the reader is referred to [18].

## REFERENCES

- [1] Fox E.A., et al. (Eds.), 1995. "Digital Libraries". [Special issue of] *Communications of the ACM* 38 (4).
- [2] NLS: National Land Survey of Finland, Opastinsilta 12 C, P.O.Box 84, 00521 Helsinki, Finland.  
[http://www.nls.fi/index\\_e.html](http://www.nls.fi/index_e.html).
- [3] Fränti P., Ageenko E., Kopylov P., Gröhn S. and Berger F., 2004. "Compression of map images for real-time applications", *Image and Vision Computing*, 22 (13), November 2004, pp. 1105-1115.
- [4] Pitas, I., Venetsanopoulos A.N., 1990. *Nonlinear digital filters: principles and applications*, Boston, Mass.: Kluwer.
- [5] Dougherty E.R., Astola J. (eds), 1997. *Nonlinear Filters for Image Processing*, SPIE Optical Engineering Press.
- [6] Serra J., 1982. *Image Analysis and Mathematical morphology*. London: Academic Press.
- [7] Matheron G., 1975. *Random Sets and Integral Geometry*, J. Wiley & Sons, New York.
- [8] Dougherty E.R., 1992, Optimal mean-square n-observation digital morphological filters. Part I: Optimal binary filters, *Computer Vision, Graphics, and Image Processing*, 55, pp. 36-54.
- [9] Heijmans H.J.A.M., 1994, *Morphological image operators*. Boston: Academic Press.
- [10] Ting D., Prasada B., 1980. Digital Processing Techniques for Encoding of Graphics. *Proceedings of IEEE*, 68 (7), pp. 757-769.
- [11] Wah, F.M., 1986, A binary image preprocessor for document quality improvement and data reduction, *Proc. Int. Conf. on Acoustic, Speech, and Signal Processing-ICASSP'86*, pp. 2459-2462.
- [12] Zhang Q., Danskin J.M., 1996. Bitmap reconstruction for document image compression. *SPIE Proc. Multimedia Storage, Archiving Systems*, Boston, MA, USA, Vol. 2916, pp.188-199.
- [13] Ping Z., Lihui C., Alex K.C., 2000. Text document filters using morphological and geometrical features of characters, *Proc. Int. Conf on Signal Processing-ICSP'00*, pp. 472-475.
- [14] Randolph T.R., Smith M.J.T., 2001. Enhancement of fax documents using a binary angular representation, *Proc. Int. Symposium on Intelligent Multimedia, Video and Signal Processing*, Hong Kong China, 2-4 May 2001, pp. 125-128.
- [15] Zheng Q., Kanungo T., 2001. Morphological degradation models and their use in document image restoration, University of Maryland, USA, Technical Report, LAMP-TR-065 CAR-TR-962 N660010028910/IIS9987944.
- [16] Ageenko E., Fränti P., 2000. Context-based filtering of document images, *Pattern Recognition Letters*, 21 (6-7), Elsevier Science, pp. 483-491.
- [17] Kolesnikov A., Fränti P., 2005. Data reduction of large vector graphics, *Pattern Recognition*, 38(3), pp 381-394.
- [18] Ageenko E. and Podlasov A., 2005, "Rapid implementation of image processing algorithms with ImageJ", *Proc. IADIS Informatics International Conference (I2005)*, within the *IADIS Virtual Multi Conference on Computer Science and Information Systems (MCCSIS 2005)*, April 11-29, to appear.