IADIS International Journal on Computer Science and Information Systems
Vol. 1, No. 1, pp. 101-114
ISSN: 1646-3692

ON THE RESTORATION OF SEMANTIC FEATURES IN RASTER TOPOGRAPHIC IMAGES

Eugene Ageenko and Alexey Podlasov
Department of Computer Science, University of Joensuu, Box 111, FIN-80101 Joensuu, Finland

ABSTRACT

Raster topographic 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 map. 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. The technique requires few computation resources and can be successfully used in mobile computers and terminals.

KEYWORDS

Topographic images, semantic, restoration, morphology.

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. Whereas maps could be effectively stored in vector format, raster imagery data is more preferable on a client-side since it is easier to transmit and handle, and does not raise any compatibility issues.

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 and can be easily stored on a server side in a database, it is significantly more resource demanding for the client to handle vector data. Therefore generated raster map images are after more preferable. Raster images are often used for digital publishing of maps on CD-ROM or in the majority cartography services 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.

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 [8-10]. 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 [11-15]. 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 contextual information [16-19]. Still most of previously developed methods are not directly applicable in 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. In our task we can not use that kind of enhancement because we made an agreement that the content of map images is critically important and thus can not be degraded.

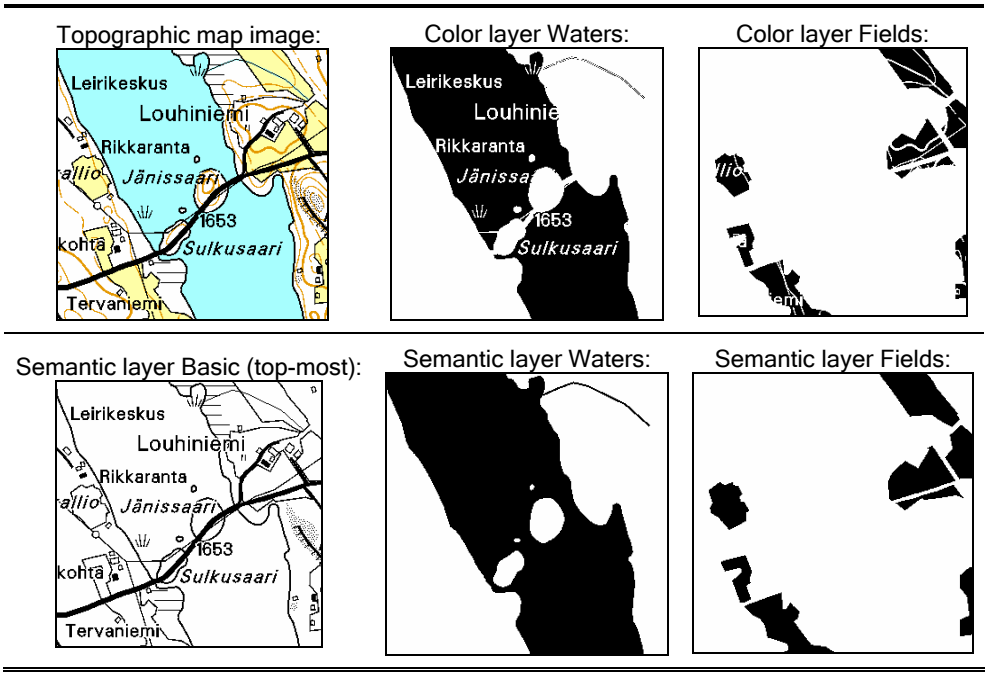


Figure 1. Illustration of map image, its semantic structure, and color layers showing the artifacts due to color separation
(with permission of National Land Survey of Finland).

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 that are degraded due to separation and would be anyway overlapped with other layers during composition. Therefore the color image obtained using combination of the reconstructed layers exactly matches the initial image without any loss of 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 $E = \mathrm{E}^2$ (the space of all possible image pixel locations), and the *binary image X* – as a set $X \subseteq E$. By $\mathrm{P}(E)$ we denote the power set of *E* comprising all subsets of *E*. The notations $X \subseteq E$ and $X \in \mathrm{P}(E)$ are equal. 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. In mathematical morphology the following operators are defined [9]:

1) *dilation of X by A*, denoted by $\delta_A(X)$, as an operator on $\mathrm{F}(E)$ such as:

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

2) *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.

3) *(structural) opening of X by A*, denoted by α_A is:

$$\alpha_A(X) = \delta_A(\varepsilon_A(X)), \tag{3}$$

4) dually, (*structural*) *closing of X by A*, denoted by β_A, is consequently:

$$\beta_A(X) = \varepsilon_A(\delta_A(X)) \tag{4}$$

5) *rank operator* $\rho_{A,r}(X)$ is the translation invariant operator that sets current pixel to be foreground if the amount of foreground pixels in a neighborhood defined by the structuring element is greater than *r* (5); otherwise the pixel is defined as a background pixel.

$$\rho_{A,r}(X) = \left\{h \in E \,\middle|\, card(X \cap A_h) \geq r\right\}, \tag{5}$$

where *card (Y)* is a cardinal value (a number of elements) of a set *Y*.

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

$$\delta_A(X) = \rho_{\tilde{A},1}(X) \text{ and } \varepsilon_A(X) = \rho_{A,card(A)}(X). \tag{6}$$

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 $\psi(A \mid T)$ is called $\psi(A)$ *conditioned relative to T* and defined as follows:

$$\psi(A \mid T) = \psi(A) \cap T. \tag{7}$$

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

2.3 Soft morphological operators

Standard, or *crisp*, morphological operators are based on local maximum and minimum operations while soft morphological operations are base on more general weighted order statistics. This makes soft morphological filters to be less sensitive to additive noise and small variations in the shapes of the object o be filtered than standard morphological filters. The definitions of the soft morphological operators are similar to crisp operators but incorporate a factor, *r*, of how well the structuring element fits within the image [9, 10].

The basis of soft morphology consists of *soft dilation* and *soft erosion* operators that are defined as follows:

$$\delta_A(X,r) = \left\{ h \middle| card\left(X \cap \tilde{A}_h\right) \geq r \right\} = \rho_{\tilde{A},r}(X), \tag{8}$$

$$\varepsilon_A(X,r) = \left\{ h \middle| card\left(X \cap A_h\right) \geq r \right\} = \rho_{A,r}(X). \tag{9}$$

The factor *r* represents the minimum acceptable overlap between *X* and the displaced structuring element *A* and regulates the amount of shrinking (in case of erosion) or expanding (in case of dilation) of objects with the image. Standard (*crisp*) dilation (1) and erosion (2) are special cases of their soft counterparts (8) and (9):

$$\delta_A(X) = \delta_A(X,1), \text{ and } \varepsilon_A(X) = \varepsilon_A\left(X, card(A)\right). \tag{10}$$

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 $\mathsf{M} \xrightarrow{D} \mathsf{M}$ of splitting *n*-color map image M into a layer set $\mathsf{M} = \{L_1 ... L_n\}$ 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:

$$\forall \mathsf{M}_1 = \{L_1, ..., L_n\} \xrightarrow{C} \mathsf{M} \xrightarrow{D} \mathsf{M}_2 = \{L_1^*, ..., L_n^*\} \subseteq \mathsf{M}_1, \tag{11}$$

where $\mathsf{M} \xrightarrow{C} \mathsf{M}$ 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

$$\begin{cases} \psi(L_k): \mathrm{P}(E) \to \mathrm{P}(E) \\ \{L_1, \cdots, L_k, \cdots L_n\} \xrightarrow{C} \mathsf{M}_1 \\ \{L_1, \cdots, \psi(L_k), \cdots L_n\} \xrightarrow{C} \mathsf{M}_2 \end{cases} \tag{12}$$

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 $\mathsf{M}_2 = \mathsf{M}_1$ 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 in the corrupted layer. It can only add pixels to a layer, so that the condition (13) is maintained:

$$L_k \subset \psi(L_k \mid T_k). \tag{13}$$

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

3.2 Layer reconstruction

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

$$\mathrm{M} = \begin{pmatrix} C \subseteq E \\ c : \mathbf{Z}^2 \to \square \end{pmatrix}, \tag{14}$$

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, respectively. Our restoration algorithm (referred further as *Iterative Semantic Layer Restoration*, or shortly *ISLR*) takes this image as an input and generates a set of reconstructed semantic layers $\hat{\mathrm{M}} = \{\hat{L}_1 ... \hat{L}_n\}$ as output. The core of the ISLR algorithm is the operator ψ such as:

$$\psi(L|T) : \begin{bmatrix} L \\ T \end{bmatrix} \to \begin{bmatrix} \delta_{A,r}(L|T) \\ \varepsilon_{B,r}(T) \cup \delta_{A,r}(L|T) \end{bmatrix}, \tag{15}$$

where L is the layer and T is its mask.

The diagram of restoration process is outlined in Figure 3 and the pseudo-code of the algorithm is shown in Figure 4. First the map image is decomposed into a set of color layers $\mathrm{M} = \{L_1 ... L_n\}$. Then for each layer the mask is constructed, and the layer restoration operator ψ is applied iteratively for each layer and its mask. At every iteration, the object areas spread within the mask, and then the mask area shrinks. The iterative process is continued until the layer and mask converges. However, examination if the mask and layer are equal is a time consuming operation, especially if the image size is big. To avoid unnecessary delays, we consider the second approach assuming that most of the artifacts have limited structural size. Therefore we consider it sufficient to perform a small predefined number (m) of iterations to complete the restoration process. Finally, the result of the iterative process is a set of reconstructed layers $\hat{\mathrm{M}} = \{\hat{L}_1 ... \hat{L}_N\}$.

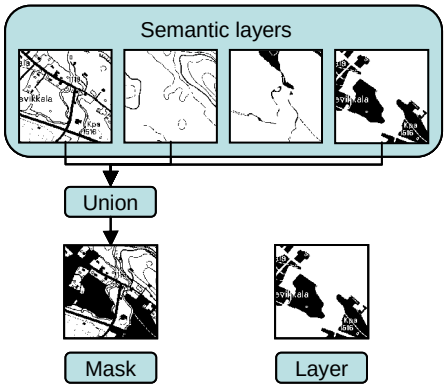


Figure 2. Block diagram of the mask construction.

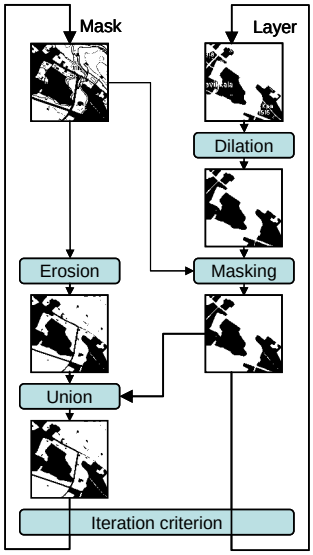


Figure 3. Block diagram of the layer restoration.

BEGIN	Input: n-color map image M

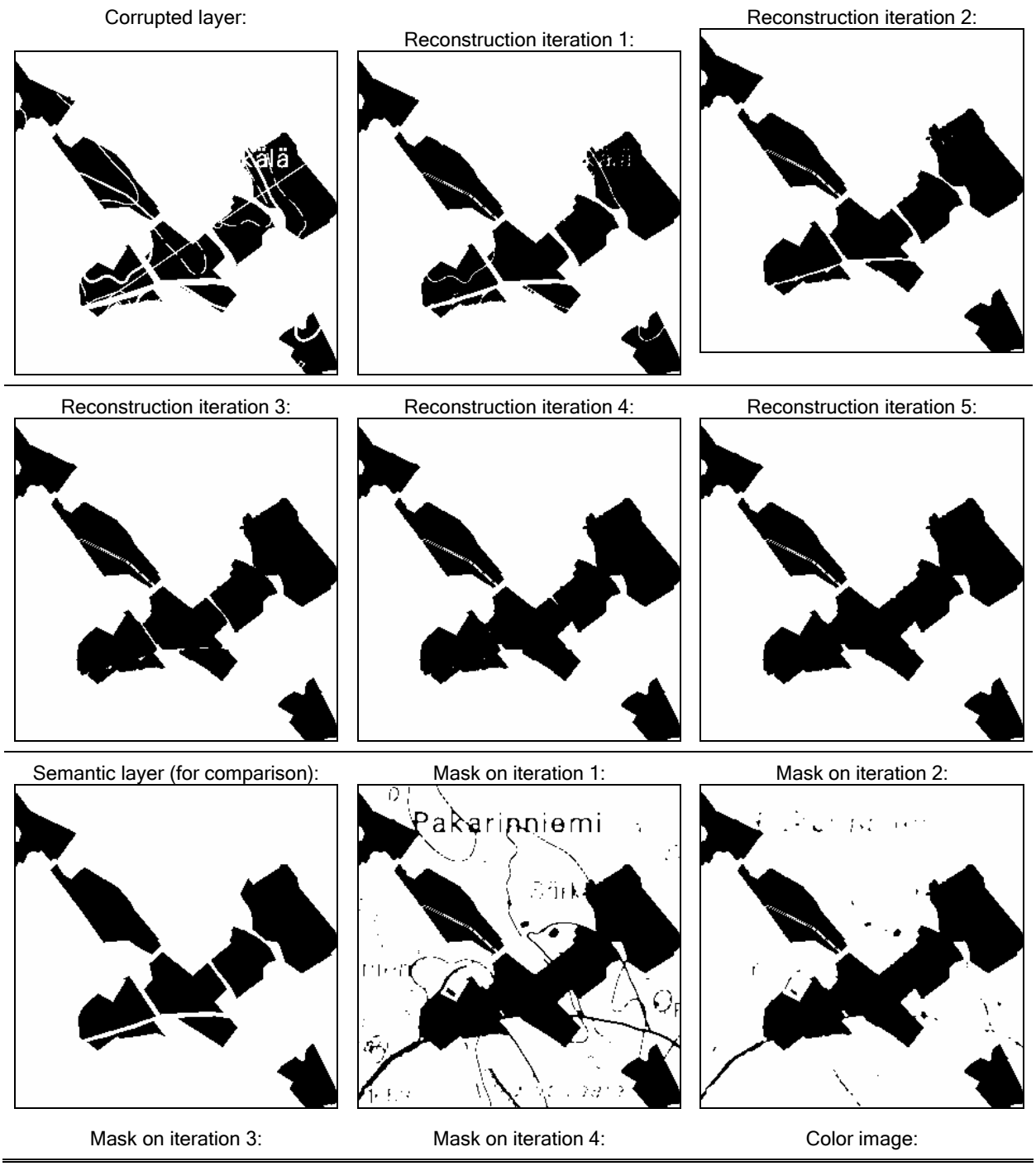
FOR $k=1$ TO N DO
$L_k=\{h\in C \mid c(h)=k\}$
$T_k=\bigcup_{j=1}^{k} L_j$
END FOR
	Preparation stage:
	layer decomposition $\mathrm{M} \xrightarrow{D} \mathsf{M}=\{L_1...L_N\}$
	mask $\mathsf{T}=\{T_1...T_N\}$ creation

FOR $i=1$ TO m
FOR $k=1$ TO n DO
$L_k^i=\delta_{A,r}\left(L_k^{i-1}\middle\vert T_k^{i-1}\right)$
$T_k^i=\varepsilon_{B,r}\left(T_k^{i-1}\right)\cup L_k^i$
END FOR
END FOR
	Restoration stage:
	Restoration operator ψ applied iteratively.
	We assume here that $L_k^0\equiv L_k$ and $\hat{L}_k\equiv L_k^m$,
	where m is the last iteration.

END	Output: $\hat{\mathsf{M}}=\{\hat{L}_1...\hat{L}_N\}$

Figure 4. Layer reconstruction algorithm.

The stepwise process of the iterations is illustrated in Figure 5, which shows the corrupted layer fragment L^0, its reconstruction iterations L^i, masks T^i for each step, and the original semantic layer, and color map image. Also, the figure illustrates very well the idea of layer overlapping. We have used the assumption that *Fields* may not overlap *Waters*, but can be overlapped by the *Basic* layers. Please observe how the field is reconstructed under the road; however in reality it has the gap there. If the roads would be separated to their own layer, this situation would have been solved.


Corrupted layer:
Reconstruction iteration 1:
Reconstruction iteration 2:
Reconstruction iteration 3:
Reconstruction iteration 4:
Reconstruction iteration 5:
Semantic layer (for comparison):
Mask on iteration 1:
Mask on iteration 2:
Pakarinniemi
Mask on iteration 3:
Mask on iteration 4:
Color image:

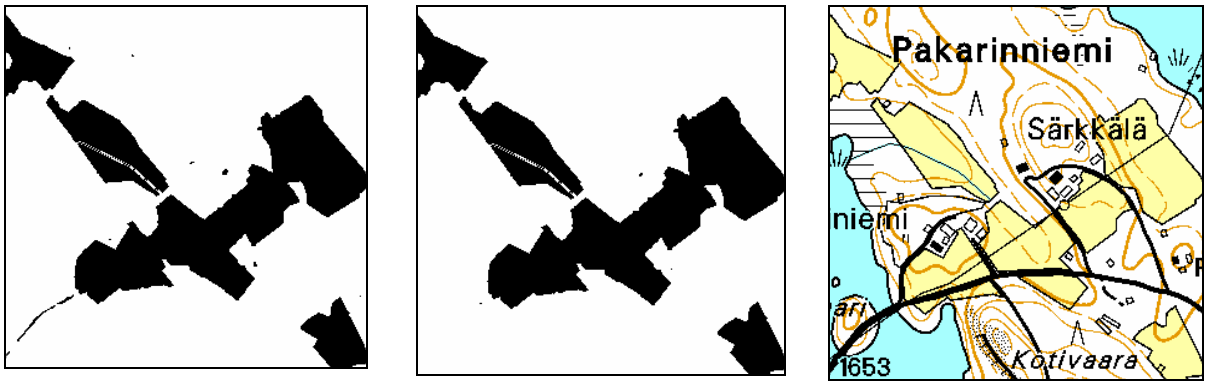


Figure 5. Iteration-by-iteration illustration of the ISLR algorithm: corrupted layer fragment, its reconstruction and mask iterations, and the semantic layer fragment. Mask for the last iteration is omitted in favor of color map fragment.

3.3 Map image regeneration

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

$$\hat{\mathrm{M}} = \begin{pmatrix} C = \bigcup_{k=1}^{n} \hat{L}_k \\ c(h) = \min_{\forall k | h \in \hat{L}_k} k \end{pmatrix}, \tag{16}$$

Due to (13) it can be easily proven that the resulting combined map image is identical to the source map image:

$$\mathrm{M} \xrightarrow{D} \mathrm{M} \xrightarrow{\psi} \hat{\mathrm{M}} \xrightarrow{C} \hat{\mathrm{M}} = \mathrm{M}\ , \tag{17}$$

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*" [2]. NLS map images are composed of four semantic layers: *Basic* topographic data, *Elevation* lines, *Water* areas and *Fields* of agriculture. Some images also contain *Boundaries* of property layer. 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 undistorted semantic counterparts. Due to the structure of the Elevation lines layer, it cannot be completely reconstructed using the proposed technique. Therefore, we have not applied the restoration for this layer.

Important parameters of the restoration algorithm are:

- appropriate structuring elements, and
- number of iterations *m*.

There are two structuring elements used in the ISLR algorithm: *A* in the layer dilation phase and *B* 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 object expansion and mask reduction. In our investigations, we have used two different structuring elements: *block 3×3*, and *cross 3×3* (the origin is marked as $\widehat{\bullet}$):

$$block = \begin{pmatrix} \bullet & \bullet & \bullet \\ \bullet & \widehat{\bullet} & \bullet \\ \bullet & \bullet & \bullet \end{pmatrix}, \text{ and } cross = \begin{pmatrix} & \bullet & \\ \bullet & \widehat{\bullet} & \bullet \\ & \bullet & \end{pmatrix} \tag{18}$$

For structuring element *block* we have applied also soft morphological dilation and erosion with factors 2 and 8 correspondingly. The entire structuring element *cross* has been used in favor of other rank factors. We have experimented with many different cases, and selected nine most prominent candidates for our evaluation. The parameters for the ILRS algorithms for these nine cases are summarized in the Table 1. The default rank factor values are not shown in the table for simplicity of reading. Default value of rank factor is 1 for the crisp dilation and *card*(*structuring element*) for crisp erosion. We have also applied several iterations of the algorithm. As future results will show, only 3 – 4 iterations are sufficient for obtaining efficient results for our test image set.

Table 1. The parameters for the layer growth and mask reduction phases of the ILSR algorithm in the form: structuring element, rank factor. Default rank factors are not displayed.

Layer growth	block	block	block	block, 2	cross	block, 2	block, 2	cross	cross
Mask reduction	block, 2	cross	block	block	block	cross	block, 8	cross	block, 8

4.1 Restoration quality – Hamming distance

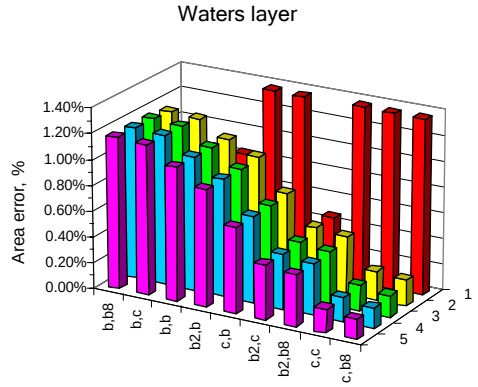
By restoration quality could be considered any particular measure that calculates how well the corrupted layers were reconstructed. The simplest way is to measure the difference between undistorted semantic layers and the reconstructed semantic layer using the proposed technique. In particular, we present 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. These values for Water and Fields layers are shown in the Table 2. All reconstruction algorithms behave rather well by reducing the Hamming distance to about 0.002 – 0.003.

Table 2: The Hamming distance between reconstructed image and semantic layer for map layers Waters and Fields. The iteration step 0 corresponds to the corrupted layers. In the table, c stands for *cross*, and b stands for *block* structuring elements.

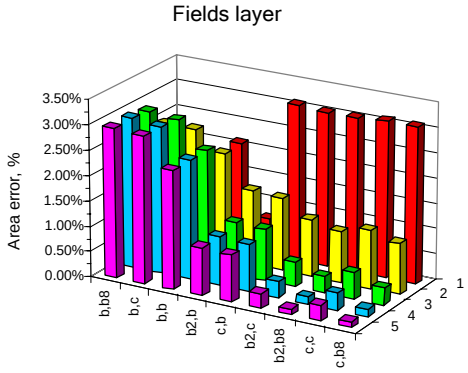
	Waters layers									**Fields layers**								
Iteration	**b,b8**	**b,c**	**b,b**	**b2,b**	**c,b**	**b2,c**	**b2,b8**	**c,c**	**c,b8**	**b,b8**	**b,c**	**b,b**	**b2,b**	**c,b**	**b2,c**	**b2,b8**	**c,c**	**c,b8**
0	1.86%	1.86%	1.86%	1.86%	1.86%	1.86%	1.86%	1.86%	1.86%	0.83%	0.83%	0.83%	0.83%	0.83%	0.83%	0.83%	0.83%	0.83%
1	0.24%	0.24%	0.24%	0.27%	0.34%	0.27%	0.27%	0.34%	0.34%	0.24%	0.24%	0.24%	0.20%	0.21%	0.20%	0.20%	0.21%	0.21%
2	0.24%	0.24%	0.23%	0.22%	0.26%	0.18%	0.18%	0.21%	0.22%	0.30%	0.29%	0.29%	0.19%	0.19%	0.19%	0.19%	0.19%	0.20%
3	0.24%	0.24%	0.23%	0.22%	0.25%	0.18%	0.18%	0.21%	0.21%	0.32%	0.31%	0.30%	0.22%	0.21%	0.22%	0.23%	0.22%	0.23%
4	0.24%	0.24%	0.23%	0.22%	0.25%	0.18%	0.18%	0.21%	0.21%	0.33%	0.32%	0.30%	0.22%	0.22%	0.23%	0.24%	0.23%	0.24%
5	0.24%	0.24%	0.23%	0.22%	0.25%	0.18%	0.18%	0.21%	0.21%	0.33%	0.32%	0.30%	0.22%	0.22%	0.23%	0.24%	0.24%	0.24%

4.2 Area measurement

In the following, we compare area measured over the original semantic layer with one measured over reconstructed color layer. The results are presented for Water and Fields layers separately on average within the test set, see Figure 6. The results show that reconstruction reduces the error of the area measurement from 12-14 % to less than 0.2 %. The results show the general trend between various parameter sets. One interesting observation is that some parameter sets give best results already on the first iteration, and then results become worse, whereas algorithm executed with other parameter sets require several iterations, but gives substantially better results. Among the parameters of the first type, we select sets {(block),(block,8)} and {(block),(cross)}, which behave equally good for all the images – error stays within 0.7–0.8 %. Among the parameters sets of the second type, we select the set {(cross),(block,8)}, which gives error rates well under 0.2 %, but requires 3–5 iterations. Thus, we end-up with two parameters set: one suitable for portable device with limited computational resources, and second one, which is about 4 times better but 3–5 times slower than the first one.



	b,b8	b,c	b,b	b2,b	c,b	b2,c	b2,b8	c,c	c,b8
1	0.83%	0.83%	0.83%	1.37%	1.36%	0.47%	1.37%	1.36%	1.36%
2	1.15%	1.14%	1.02%	0.93%	0.69%	0.47%	0.45%	0.22%	0.20%
3	1.17%	1.16%	1.04%	0.91%	0.67%	0.44%	0.41%	0.19%	0.17%
4	1.17%	1.16%	1.04%	0.91%	0.67%	0.43%	0.40%	0.19%	0.16%
5	1.18%	1.16%	1.04%	0.91%	0.67%	0.43%	0.40%	0.19%	0.16%



	b,b8	b,c	b,b	b2,b	c,b	b2,c	b2,b8	c,c	c,b8
1	0.71%	0.71%	2.11%	0.71%	3.10%	3.05%	3.05%	3.10%	3.10%
2	2.49%	2.47%	2.11%	1.48%	1.45%	1.13%	1.02%	1.15%	1.01%
3	2.92%	2.86%	2.36%	1.03%	1.02%	0.48%	0.33%	0.52%	0.34%
4	2.96%	2.91%	2.36%	0.96%	0.93%	0.31%	0.14%	0.34%	0.14%
5	2.97%	2.91%	2.36%	0.95%	0.92%	0.29%	0.11%	0.31%	0.11%

Figure 6. The error of area measurement procedure for Waters and Fields layers. The results are given for nine parameter sets and five iterations.

To further demonstrate the usability of the method, we compare the results with the reconstruction based on non-iterative operator (17) referred further as *conditional closing* ς, where C is a 7×7 pixel block:

$$\varsigma_{T_k}(L_k): L_k \to \beta_C(L_k \mid T_k) \tag{19}$$

Since ISLR approximates layers better than CC, its area measurements are also much closer to the semantic layers than the CC results. The Table 3 summarizes the results for CC and ISLR algorithms; the latter is used with best parameter set.

Table 3. The area (in square meters) measured over original semantic layers, color layers, and reconstructed semantic layers with CC and ISLR (c,b8) algorithms for Water and Fields layers.

Topographic layer	Semantic layers	Color layers		Reconstructed ς (CC)		Reconstructed ψ (ISLR)	
	Area	Area	Error, %	Area	Error, %	Area	%
Waters	915,753	799,233	12.72	873,628	4.6	914,313	0.16
Fields	365,148	313,499	14.14	347,255	4.9	364,755	0.11

4.3 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 ISLR restoration technique 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 7 illustrates the visual effect of the layer removal.

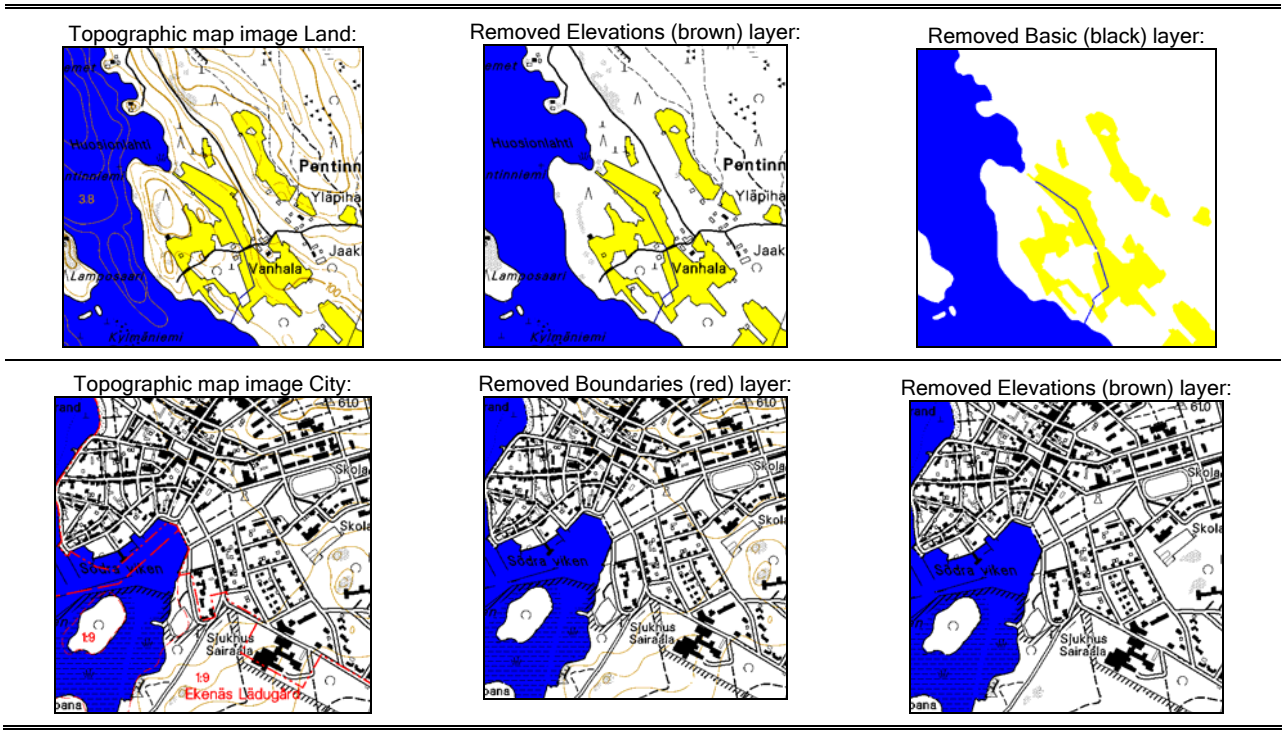


Figure 7. Example of the layer removal for two map images: Land and City. First column shows original raster map image fragments, second column – same fragment with one layer removed, and third – with two layers removed.

4.4 Layer compression

Even though it has not been our primary concern, we have also examined how the restoration procedure affects the compression performance of the reconstructed layers. We have evaluated the ILSR algorithm against three compression techniques, namely LZ (PNG), ITU Group 4 (TIFF) and JBIG. At all times we have obtained 10-30% improvement with LZ and 30-55% with G4 and JBIG. Please note that even though such high compression performance was achieved for reconstructed layers, it will have only a marginal impact on the map image in total. This is due to vast amount of information containing in the Basic layer and therefore its much larger file size even in the compressed form.

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 14 % to less than 0.2 %. The reconstructed layers have lesser entropy and can substitute for the color layers in map data storage without any loss of quality, which subsequently improve compression performance. 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 mobile terminals.

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.

[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] Koskinen L., Astola J., 1994. Soft morphological filters: a robust morphological filtering method, *Jurnal of Electronic Imaging*, 3, pp. 60-70.

[11] Ting D., Prasada B., 1980. Digital Processing Techniques for Encoding of Graphics. *Proceedings of IEEE*, 68 (7), pp. 757-769.

[12] 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.

[13] 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.

[14] 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.

[15] Randolph T.R., Smith M.J.T., 2001. Enhancement of fax documents using a binary angular representation, *Proc. Int. Symp. on Intelligent Multimedia, Video and Signal Processing*, Hong Kong China, 2-4 May 2001, pp. 125-128.

[16] 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.

[17] Ageenko E., Fränti P., 2000. Context-based filtering of document images, *Pattern Recognition Letters*, 21 (6-7), Elsevier Science, pp, 483-491.

[18] Kolesnikov A., Fränti P., 2005. Data reduction of large vector graphics, *Pattern Recognition*, 38(3), pp 381-394.

[19] Doughberty E., Lotufo R., 2003. *Hands-on morphological image processing*. SPIE Optical Engineering Press.