

# Multi-layer Filtering Approach for Map Images

Minjie Chen<sup>1</sup>, Mantao Xu<sup>1</sup>, Pasi Fränti<sup>1,2</sup>

1. University of Joensuu, Finland; 2.Fudan University, Shanghai, China

## ABSTRACT

Raster map image is an important set of color images with similar patterns and textures presented in a limited number of colors. Manipulating this class of color images using conventional image filters may lead to a severe over-filtering problem, by which important details structures are over eliminated or degraded. Even if the statistical based algorithms have been recognized as a set of most efficient filters, their exponentially high memory consumption and computational cost can make them intractable in practice. To solve this operational difficulty, this work proposed a novel multi-layer image filtering approach transforming map image filtering into binary domain. It consists of three intuitive image operators: layer decomposition, binary image filters and layer merging. Experimental results demonstrated that the new proposed approach is very efficient for filtering of raster map image.

**Index Terms**—Digital image processing, filtering.

## 1. INTRODUCTION

A class of widely used color images in geosciences is raster map images that are encoded in a regular grid of pixel colors arrayed in rows and columns. In contrast to the conventional color images (e.g., photographic images), raster map images do not allow smooth continuous tone changes in colors but merely display pixel level repetitive detail structures, which are essential for interpreting map objects. This is because each color in the image represents a different semantic map object. However, in many applications, the color information of map images may be distorted by lossy compressions, vector-to-raster conversion, or during the digitization process itself. This color distortion may lead to a severe false recognition of map objects.

A common way to address this problem is to apply image filtering before further processing. The main challenge in color image filtering is that most of color images contain multivariate data with color correlations, which make the design of image filters very difficult in practice. A proven set of intuitive color image filters is to apply gray-scale image filters for each channel of color images separately. However, this approach does not take into account the necessary color correlations, which results in a production of false colors and edge degradation. To overcome this difficulty, many filters have been studied using nonlinear

approaches. For instance, a class of efficient color image filters can be designed using a set of ordering criterions or order-statistics for color vectors [3]. However, in most cases, the set of ordering criterions must be defined in terms of the distance between two color vectors, which is usually not appropriate for our case.

This problem can be partly revealed by applying the ordering criterions only to the pixels that are identified (or assumed) as noises or outliers by using *adaptive vector median* [2] or *fast peer group filter* [4]. However, nonlinear filters designed for conventional color images may eliminate the most useful edge information (e.g., thin edges containing important information) when they are applied to map images. This is because each color in map images represents a distinct class of map objects, which forces the conventional filters to over-smooth the more detailed structures. Even though statistical modeling approaches [8, 9] have made significant progress by learning image structure and preserving the repetitive structures, their memory consumption and computational expense are exponentially high.

In this paper, we propose a multi-layer image filtering approach. Instead of using an order-statistics filter for color vectors (e.g., the adaptive vector median filter), the image  $I$  is first decomposed into a series of binary layers. For a given pixel  $(i,j)$ , amongst  $N$  number of layers of  $\{L_t(i,j)| t = 1, \dots, N\}$  there exists only one layer  $t$  where the pixel belongs to:  $L_t(i,j) = 1$ . Then each binary layer is processed separately by a binary image filtering method. In this way, we avoid the problem of dealing with color distances. Instead, the problem is reduced back into a simpler (binary) domain of which set operations (as in morphological filters), or rank-order operations can be applied.

For the sake of reconstructing the resulting filtered color image  $I^*$  from the set of binary layers  $\{L_t^*| t = 1, \dots, N\}$ , the layers must be ordered according to the color priority. Namely, for a given pixel  $(i,j)$ , its output color in the filtered image,  $I_G(i,j)$  must be selected from those layers such that  $\{L_t^*(i,j)| L_t^*(i,j) \neq 0, t = 1, \dots, N\}$ . For this reason, prioritizing the colors is needed. We apply region-based ordering where the image is first divided into segments having different background color, and then deriving the color propriety for each segment locally.

## 2. MULTI-LAYER DECOMPOSITION OF IMAGES

The use of multi-layer decomposition was originally proposed in [5] for image compression. We extend this

framework to the filtering of map image. The basic idea behind the multi-layer approach is illustrated in Fig. 1. The image decomposition is straightforward, and the following filtering can be performed using any binary image filtering method. The last step is to merge the layers back to a color image, which requires some attention.

Here,  $(i,j)$  denotes a pixel of a image  $I$  with limited number of colors, i.e.  $N \ll 256$ . The output color for the pixel,  $I(i,j)$ , can be uniquely determined by a binary vector,  $\mathbf{x}_L$ ,

$$\mathbf{x}_L = (L_1(i,j), \dots, L_N(i,j)) \quad (1)$$

where each component  $x_k$  such that:

$$L_t(i,j) = \begin{cases} 1, & \text{if } t = I(i,j) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Here,  $L_t(i,j)$  can be treated as the value of pixel  $(i,j)$  for  $t^{\text{th}}$  binary layer. Clearly, any filter applicable to binary images can be performed on each of the  $N$  number of binary layers, e.g., median filter, morphological filter [7] and the statistical filter [8]. Once each of binary layers is processed, the resulting binary layers,  $\{L^*_t \mid t = 1, \dots, N\}$ , are merged to reconstruct the output color image,  $I_G$ . However, for a given pixel  $(i,j)$ , there might be several layers  $L_C$  where the pixel is set to 1:  $\{L^*_t \mid L^*_t(i,j) = 1, 1 \leq t \leq N\}$ . Since the resulting only one color can be assigned to each pixel, it must be selected from the set  $L_C$  according to some criterion. We make the selection based on color priority so that the output color is selected as the color layer with the highest priority:

$$I_G(i,j) = \begin{cases} \arg \max_t \sum_t L^*_t(i,j) = 1 \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

In other words, the decomposed binary layers must be ordered for the sake of merging the filtered binary images using (3). In image compression, graph based algorithm [10] has been applied to find optimal ordering of the binary layers using minimum spanning tree algorithm based on a compression cost matrix. However, the construction of the cost matrix have huge computational cost, which may not be feasible for filtering, and our criterion should be based on maximizing information in the image but not minimizing it.

One useful heuristic is to order the layers according the frequency of object pixels appearing on it. The higher is the occurrence of a color, the lower priority is assigned to it. The idea is that frequent colors represent either background or other large objects, whereas infrequent colors represent finer details, which are more likely to be more important for the quality of the image. The ordering of the colors (binary layers) is the same for every pixel, and is therefore denoted as global layer ordering. An example of this is shown in Fig. 1. The drawback of this approach is that the importance of the colors can be different in different regions. For example, the dominant (background) color that should be given the lowest priority is usually white in the map shown in Fig. 1, but in water areas it is blue and in some places it can be yellow (fields). To overcome this problem, we propose a

region based algorithm for ordering the binary layers to localize the choice of color priority.

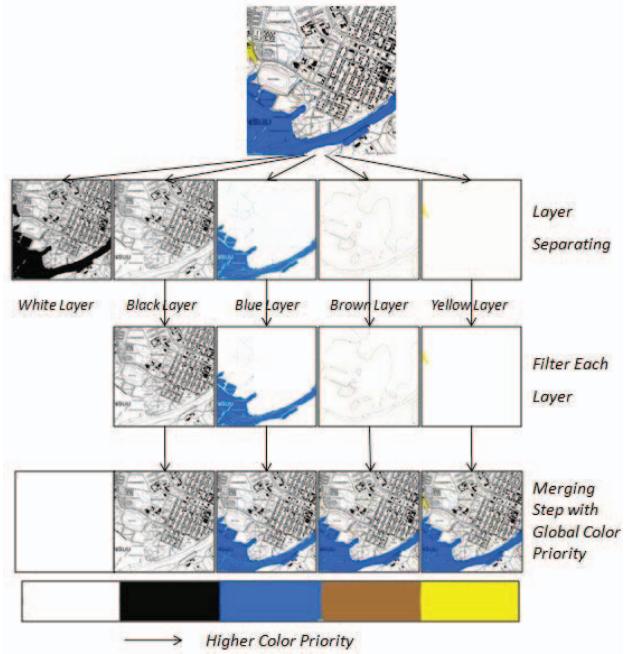


Fig. 1 Multi-layer framework using global layer ordering

### 3. REGION BASED ORDERING OF THE LAYERS

Merging the filtered binary images using global ordering results in a severe problem of damaging important disconnected structures. For example, the island inside the lake region in Fig. 1 disappeared as a result of the merging because the global layer ordering scheme assigns the *blue* color with higher priority than the *white* color, thus destroys the small island.

A region based ordering scheme is proposed for merging the binary layers  $L^*$  as follows. A multi-layer based image segmentation operation was firstly conducted to segment the raster map image into several distinct regions,  $S = \{s_k \mid \cup s_k = I, k = 1, \dots, K\}$ , and  $s_k \cap s_l = \emptyset, \forall 1 < k, l < K\}$ . After this preliminary segmentation, the color priority for each image pixel  $(i,j)$  is calculated according to the color occurrence in its region  $S_k$ , where  $(i,j) \in S_k$ . The main idea of exploiting this region based color ordering scheme is to incorporate the statistical features behind those disconnected semantic regions into image filters. For the sake of image segmentation, the conventional gradient-based segmentation algorithms in [6] are seemingly not applicable due to an inadequate number of colors for extracting the edge information. Hence, instead of performing image segmentation on the input raster map, a multi-layer based image segmentation algorithm is applied. After a dilation, hole fillings and region labeling operation on all the filtered binary layers  $L^*$ , a set of large-sized candidate or initial regions ( $M$  number of candidate regions) are extracted from all these layers. For simplicity of implementation, those

refined candidate regions are maintained in a list of size  $M$ ,  $R = \{R_m |, m = 1, \dots, M\}$  in a descending order according to their pixel size. These candidate regions in the list are evaluated one by one according to two region based features ( $f_1$  and  $f_2$ ). The first feature ( $f_1$ ) is the number of object pixels in the candidate region relative to the size of the region:

$$f_1(R_m) = \frac{\sum_{(i,j) \in R_m} L^*(i,j)}{|R_m|} \quad (4)$$

where  $R_m$  is the candidate region in the  $t(m)^{\text{th}}$  layer. The second feature ( $f_2$ ) is the percentage of labeled pixels in  $R_m$  so far:

$$f_2(R_m) = \frac{|S_M(i,j) \neq 0 \cap (i,j) \in R_m|}{|R_m|} \quad (5)$$

where  $S_M$  is a label mask image for the image segmentation  $S$  such that  $s_k = \{(i,j) | S_M(i,j) = k, (i,j) \in I\}$ . The pseudocodes of the underlying image segmentation algorithm can be found in Fig. 3.

Once the label mask image has been obtained using the algorithm in Fig. 3, the remaining non-labeled pixels ( $S_M(i,j)=0$ ) are processed by distance transform algorithm. It assigns these pixels the same label as their nearest segmented region.

For the final region  $s_k$  in  $S$ , its color frequency is calculated, and the local color ordering within this region is determined by sorting the color from lowest to highest frequency. A demonstration of the segmentation results is given in Fig. 2, by showing the background color (the lowest priority color) of each segmented region.

The overall algorithm is demonstrated in Fig. 4 by showing the resulting regions, and their corresponding color priorities. The merging process is also demonstrated where the image is composed step-by-step. At first step, the background colors are added for each region, and the remaining colors then one by one.



**Fig. 2** Background color for each segmented region

```

INPUT  $R \leftarrow$  the  $M$ -sized list of candidate regions
       $L^* \leftarrow$  the  $N$  layers of filtered binary images
OUTPUT:  $S_M \leftarrow$  label mask image
FUNCTION MultiLayerSegmentation ( $R^*, L^*$ ) RETURN  $S_M$ 
 $S_M \leftarrow 0;$ 
 $r \leftarrow 1;$ 
FOR  $m = 1$  to  $M$ 

```

```

 $f_1 \leftarrow$  calculate ratio of object pixel according to (4)
 $f_2 \leftarrow$  calculate percentage of labeled pixel according to (5)
IF AcceptanceCriterion ( $f_1, f_2$ ) is TRUE
  FOR every pixel  $(i,j) \in R_m$ 
     $S_M(i,j) \leftarrow r$ 
  END FOR
   $r \leftarrow r + 1$ 
END IF
END FOR

PROCEDURE AcceptanceCriterion ( $f_1, f_2$ ) REUTRN  $newSeg$ 
 $newSeg \leftarrow \text{FALSE}$ 
IF  $f_1 > 0.52$  and  $f_2 < 0.001$ 
   $newSeg \leftarrow \text{TRUE}$ 
ELSE IF  $f_1 > 0.6$  and  $f_2 < 0.005$ 
   $newSeg \leftarrow \text{TRUE}$ 
ELSE IF  $f_1 > 0.8$ 
   $newSeg \leftarrow \text{TRUE}$ 
ELSE
   $newSeg \leftarrow \text{FALSE}$ 
END

```

**Fig. 3** Pseudocodes of multi-layer image segmentation algorithm

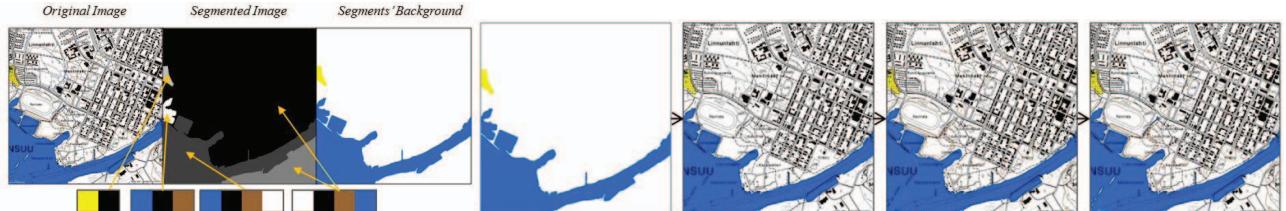
#### 4. RESULT AND DISCUSSION

We have evaluated the proposed multi-layer filtering algorithm using a set of six map images from the database provided by [1]. These images are presented by 5-16 colors, and they are of different spatial resolutions. Some of them include quantization noise whereas others are converted from vector origin. For evaluating the performance of the image filters, we artificially distort those images by adding impulsive and content-dependent noise as described in [9].

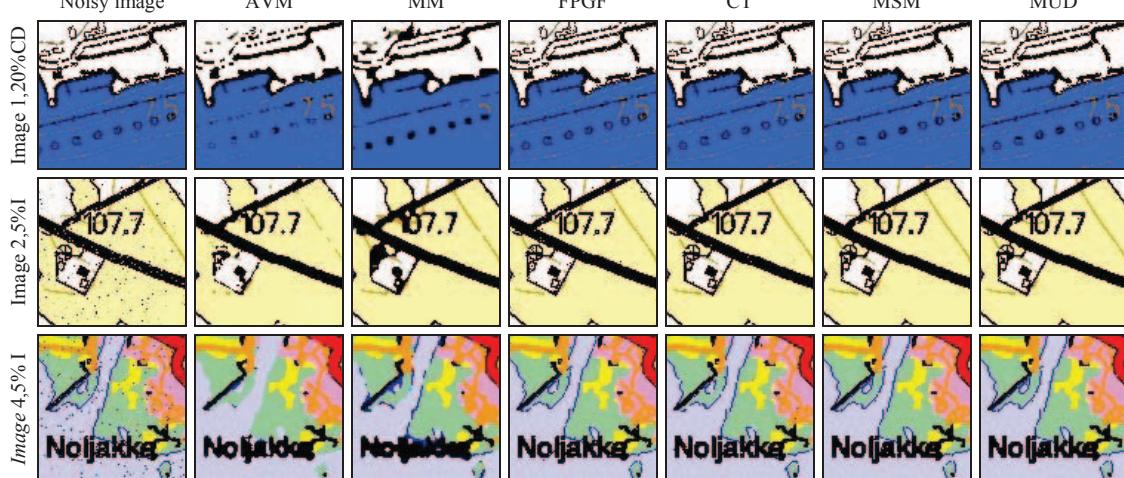
As a performance comparison, four alternative filters for color images are studied. *Adaptive vector median* (AVM) [2] and *color morphological* (MM) [3] are the filters designed for general color image filtering, while *fast peer group filter* (FPGF) [4] and *context-tree filter* (CT) [9] are tailored methods for color-index image. We measure the mean color distance ( $\Delta E$ ) in  $L^*a^*b$  color space. Although this measure does not match completely how human see the image, it gives rough idea about the relative performance of the filters.

For implementation of the proposed multi-layer image filtering approach, two binary filters, *spatially-variant morphology* [7] and *discrete universal denoising* [8] are incorporated into the framework of filtering each decomposed binary layers. We termed them as MSM and MUD in Table 1 respectively.

The objective experiment results obtained by using the six raster map image filters are summarized reported in Table 1 and subjective comparison in Fig. 5. It turns out from the results that the proposed multi-layer method can work more efficiently than the other four filters when image is converted from vector origin with less color number (Image#1, 2). When those map images are generated by quantization of scanning maps with more colors, some meaningless layers may exist because of the inaccurate quantization, which make the frequency-based ordering does not work.



**Fig. 4** An example of merging of filtered binary images (right) using the region based layer ordering (left)



**Fig. 5** Performance comparison of the six image filters tested in the experiments

**Table 1** the filter efficiency for 5% impulsive noise (I) and 20% content-dependent noise (CD)

$\Delta E$	Image	1	2	3	4	5	6
	Colors	5	5	9	10	16	16
I	AVM	3.47	5.81	8.98	7.83	1.68	6.18
	MM	4.80	4.16	12.0	10.0	2.60	12.8
	FPGF	0.95	1.00	1.77	1.45	0.87	3.35
	CT	0.77	0.95	1.54	1.97	0.83	2.24
	MSM	0.99	1.21	3.10	2.56	1.67	8.09
	MUD	0.68	0.89	1.87	1.69	1.06	3.88
CD	AVM	4.14	5.58	9.26	8.07	1.89	7.21
	MM	4.61	4.31	12.0	9.96	2.73	12.5
	FPGF	1.99	1.70	3.51	3.35	1.42	5.02
	CT	2.08	1.91	3.15	3.66	1.17	3.56
	MSM	2.03	2.02	4.52	4.05	2.05	9.01
	MUD	2.04	1.81	3.37	3.23	1.39	5.22

## 5. CONCLUSION

We have proposed a multi-layer approach for filtering algorithm for raster map images. This proposed method provided a solution for processing map image in binary domain. It also has lower computation cost and memory consumption comparing to statistical method. Experimental results have validated that the proposed algorithm is very efficient for filtering raster map images.

Future work can be done in two aspects: refining the proposed filtering algorithm in denoising the true color

images and optimizing the region based layer ordering using shape analysis.

## 6. REFERENCES

- [1] National Land Survey of Finland, (<http://www.nls.fi>).
- [2] R. Lukac, "Adaptive vector median filtering", *Pattern Recognition Letters* 24, pp. 1889-1899, 2003.
- [3] G. Louverdis, M. Vardavoulia, I. Andreadis, P. Tsalides, "A new approach to morphological color image processing", *Pattern Recognition* 35(8), pp. 1733-1741, 2002.
- [4] B. Smolka, A. Chydzinski, "Fast detection and impulsive noise removal in color images", *Real-Time Imaging*, 11, pp.389-402, 2005.
- [5] S. Forchhammer, O. Jensen, "Content Layer Progressive Coding of Digital Maps", *IEEE Trans. on Image Processing*, 11(12), pp.1349-1356, 2002.
- [6] J. Angulo, J. Serra, "Modelling and segmentation of colour images in polar representations", *Image and Vision Computing* 25, pp. 475-495, 2007.
- [7] N. Bouaynaya, M. Charif-Cherchaoui, D. Schonfeld, "Theoretical Foundations of Spatially-Variant Mathematical Morphology Part I: Binary Images", *IEEE Trans. on Pattern Anal. and Machine Intelligence*, 30(5), pp.823-836, 2008.
- [8] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdru, and M. Weinberger, "Universal discrete denoising: Known channel," *IEEE Trans. Inform. Theory*, 51(1), pp.5-28, 2005.
- [9] P. Kopylov and P. Fräntti, "Filtering of color map images by context tree modeling", *IEEE Int. Conf. on Image Processing (ICIP'04)*, Singapore, vol. 1, pp. 267-270, 2004.
- [10] P. Kopylov and P. Fräntti, "Compression of map images by multilayer context tree modeling", *IEEE Trans. on Image Processing*, 14(1), pp.1-11, 2005.