Statistical filtering of raster map images using a context tree model

Alexey Podlasov*, Pavel Kopylov and Pasi Fränti*
*Speech and Image Processing Unit, Department of Computer Science and Statistics,
University of Joensuu, P.O Box-111, 80101, Joensuu, Finland
{apodla,franti}@cs.joensuu.fi

Abstract

We propose a statistical filter using a context tree modeling. The idea of context tree is to perform selective context expansion including only those pixel combinations that really appear in the image. This makes it possible to use much larger spatial neighborhood. The proposed context tree filtering is evaluated for a set of indexed-color raster map images corrupted with generated impulsive and content-dependent noise. The objective evaluation shows improvement of 15% for content-dependent noise and up to 30% for impulsive noise comparing to the closest competitor. Visual comparisons show that the spatial structures are preserved better than by vector median, morphological and peer group averaging filter.

1. Introduction

Geographical map images are typically present in two fundamentally different formats: raster and vector. Vector format is more suitable for large databases providing excellent flexibility and compression even though vector processing can be computationally expensive. Raster images are easier to process and this format is more suitable for final client-side processing for delivery, local archive storage and web-publishing. Typically, vector-to-raster conversion does not affect the quality of the raster image presented to the user. However, cases when the original vector data is not available are common. Raster image can be degraded by noise caused by transformations and lossy compression. Distortion also appears when a printed map is digitized. In these cases, the presence of noise can corrupt the spatial structures in the image.

A great variety of noise removal techniques are known for color image processing [1][2][3]. However, map images require some restrictions to be set. Firstly, the image should not be smoothed and it should remain readable. Secondly, the number of colors is typically small in a map image and

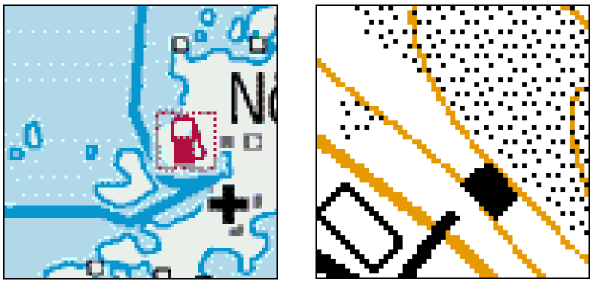

Figure 1. Examples of complicated structures that are treated as noise by most filters.

preferably it should not be increased. Thirdly, the spatial structures in the image should be preserved since they have distinctive meaning. Linear filtering methods cannot be effectively applied to map images because of their smoothing effect, which cannot be tolerated in map images. Among popular non-linear filtering there are methods such as *morphological filters* [4]; *directional vector filters* [5]; a class of *weighted median filters* [6]; its vector extension referred as *vector median filter* (VM) [7] and the adaptive variant referred as *adaptive vector median filter* (AVM) [8]. *Peer group analysis* (PGA) [9] is an edge-preserving smoothing technique based on finding a group of pixels similar to the current one in a local neighborhood. In case there is such group, the pixel is replaced with the average of its peer group.

However, existing filtering methods are mostly designed for continuous-tone images and they do not apply well for map images, web graphics and similar. This kind of images include complicated spatial structures such as one-pixel thin lines, textured areas, dashed and dotted lines, text and symbols. False filtering of this kind of structures is typical for most filters designed for photographic imagery since they tend to consider noise as a local intensity variation without taking into consideration the repeated patterns in the globally in the image. On the other hand, high variance does not necessarily identify the noise. The regions with written text or textured background are far from being uniform but their presence is vital for the usability of the map.

The examples of such structures are illustrated in

Figure 1. The area consisting of isolated black pixels on the map represents sand field in nature. Single pixels and thin lines are considered as noise by most of the existing filters, and thus, they are filtered erasing important geographical information. Morphological filtering would be a natural choice to consider for this kind of imagery. However, the drawback of morphological filtering is the concept of *structural element*, defining the preferred configuration of local patterns where domination of some pixels over the others is emphasized. It is clear that the variety of patterns in a map image is great and one or even a set of structural elements is not able to describe it accurately. Moreover, color morphology is a generalization of gray-scale morphology made by *reduced ordering*, i.e. the 'domination' relationship is defined on color vectors analogously to gray-scale intensity values. However, it seems that in color map images no color can be considered prevailing over the others just by its vector characteristics like energy and intensity.

In this paper, we introduce a statistical filter based on conditional probability estimation allowing the preservation of detailed structures in map images. The proposed filter consists of two stages: *analysis* and *filtering* stage. In the analysis stage, local conditional probabilities are estimated within the image by gathering statistics of how often each particular color appears within the same local neighborhood, called *context*. The size of the context is then optimized by using a *context tree*. The analysis stage does not consider any *a priori* knowledge about the imposed noise characteristics. In the filtering stage, all pixels that have color of low probability in its context, are considered as noise and replaced by the most probable color. In this way, the repetition of local patterns can be discovered within the image. Patterns that appear frequently enough are considered belonging to the image structure and preserved. Pixels that are unlikely to appear in their neighborhood are considered to be noise and filtered out. This property allows the filter to preserve borders and structures independently of their size or variance. Preliminary version of the work has been presented in [10]. Similar filter was considered in [11][12], where context modeling and filtering decision is made in assumption that probabilistic characteristics of noisy channel are known.

The rest of the paper is organized as follows: the proposed filter is described in Section 2; noise models are considered in Section 3; the results of experiments are presented in Section 4; and conclusions drawn in Section 5.

2. Context Tree filter

2.1 Context-based statistical filtering

Consider an image I as a rectangular grid of pixels $I(x,y)$, where (x,y) is a position of a pixel and $I(x,y)$ is its value, or color. Let $I(x,y) \in \{1, \ldots, k\}$, $\forall(x,y)$, where k is the *number of colors* in the image. We assume that k is small enough to allow the storage of the image in palette-indexed format. We define a *context* $c = \{ I(x_1,y_1), \ldots, I(x_n,y_n) \}$ as a set of n pixels, where n is denoted as the a size of the context c. The positions of the pixels in a context $(x_1,y_1), \ldots, (x_n,y_n)$ are defined as a set of offsets to the position of the current pixel, and is referred as a *context template*. In Figure 2, A illustrates a sample 20-pixel context template where the position of the current pixel is marked with '×'. The context B illustrates a sample context for a binary case, *i.e.* $I(x,y) \in \{background, foreground\}$, $\forall(x,y)$, and C illustrates similar example with more colors available.

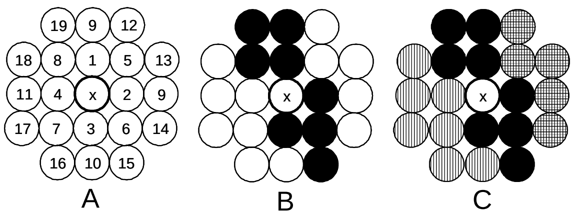

Figure 2. Template used by context tree (A) and sample contexts for the case of binary (B) and color (C) images.

The context defines the configuration of neighboring pixels and the same configuration can repeat in the image on different positions. When the neighborhood of the current pixel $I(x,y)$ equals to a context c we say that pixel $I(x,y)$ appears in a context c, and denote it as $I(x,y) \in c$. Note that the current pixel value is excluded from the context, meaning that different pixel values can appear in the same context. We associate each context c with a vector $p^c = (p^c{}_1, \ldots, p^c{}_k)$ called a *vector of statistics*, where $p^c{}_i$ represents a number of times the pixel of color i appeared in a context c in the image. After the vectors of statistics have been gathered for every context of the image, the conditional probability of every pixel to appear in its context can be estimated as

$$p(I(x,y) = j \mid I(x,y) \in c) = \frac{p^c{}_j}{\sum_{i=1..k} p^c{}_i} \qquad (1)$$

We denote this probability as $p(I(x,y)|c)$.

After the statistics have been gathered, the actual filtering is performed requiring a separate pass over the image. The main idea of the proposed filter is based on the assumption of statistical consistency of the image data. We expect that patterns appear in the image frequently enough, *i.e.* conditional probability $p(I(x,y)|c)$ of a pixel is higher than a predefined threshold for most of the pixels. Otherwise, the pixel is considered as noise and filtered out. As a replacement strategy we consider to replace the noisy pixels with the most probable color in the context. Formally, the algorithm can be outlined as follows:

```
  Analysis stage:
  For each (x,y) do
    C = {I(x1,y1), …, I(xn,yn)};
    p^c_I(x,y) = p^c_I(x,y) +1;
  For each C do
Calculate P(I=j|C) ∀j∈[1,k] as (1)

  Filtering stage:
  For each (x,y) do
    If p(I(x,y)|c) < Threshold
```

$$I(x,y) = \arg\max_{j=1..k}\left(p(I(x,y)=j \mid I(x,y)\in c)\right)$$

The concept is illustrated in Figure 3 for image consisting of three unique colors. For simplicity we consider context tree filtering within 3×3 neighborhood, and two sample contexts (A and B). In the same context, some pixel values are less probable than the others, *e.g.* black pixel is much less likely to appear than white pixel in Context A, and vice versa, white pixel is much less probable than black pixel in Context B. The probability of these pixels falls below the threshold, and therefore, the pixels are filtered by replacing with the values of the most probable ones. Three examples of contexts and their corresponding probability distributions obtained in experiments with 5-color images are presented in Figure 4. There is a clear domination of the most probable color over the others.

2.2 Context Tree modeling

Gathering pixel occurrence statistics requires one pass over the image and allocating memory for as much as there are different contexts in the image. This number is upper bounded by the number of pixels in the image. In order to optimize the memory allocation we organize the storage of statistics as a tree structure called *context tree* (CT). Similar structures have been used for probability estimation in binary image compression [13] and indexed color image compression [14].

In context tree, a context is sequentially constructed pixel-by-pixel, or to say more precisely, position-by-position according to a predefined ordered context template such as the one in Figure 2.

Each node stores a vector of statistics for its context: f_W for the number of white pixels and f_B for

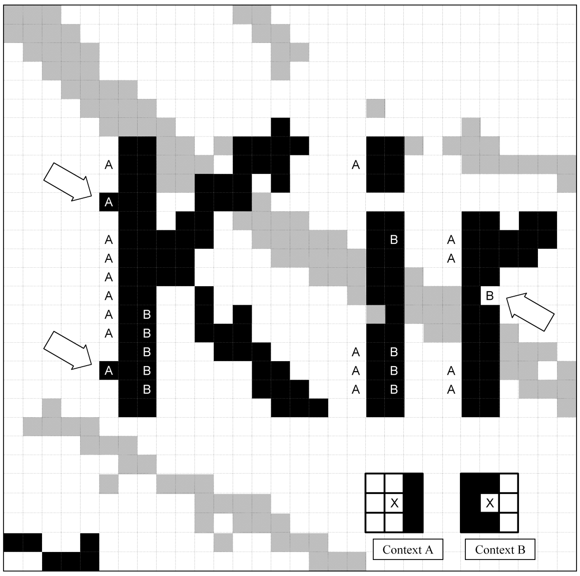

Figure 3. Example of statistical filtering. Two sample contexts are marked by A and B. The filtered less probable pixels are pointed by arrows.

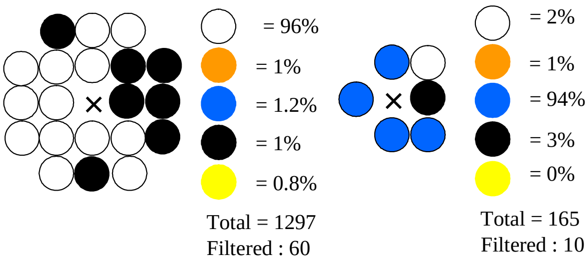

Figure 4. Sample contexts and the statistical distribution of the colors in a 5-color map image.

the number of black pixels in Figure 5. Statistics are gathered only for those contexts that really appear in the image. The principle is illustrated in Figure 5 and Figure 6 for the case of binary and a 4-color images, respectively. Every node of the tree represents a particular combination of the template pixels.

The deeper the tree grows the larger context model is used. Usually the image is processed pixel-by-pixel. For every pixel, the tree is traversed down to the desired depth, and by updating all pixel counters for the corresponding nodes along the path from the root to a leaf. When a context appears first time in the image and the corresponding node tree does not exist in the context tree, it must then be created dynamically at this moment.

Potentially, the final level of the tree can contain k^n nodes, where n is the size of the template. However, since not all possible contexts are present in the image, some nodes will never be constructed and, therefore, memory will be allocated only for existing pixel combinations. For the case of color image (see Figure 6), the construction of the tree proceeds in the same manner as in the case of binary image, expect that there can potentially be as many

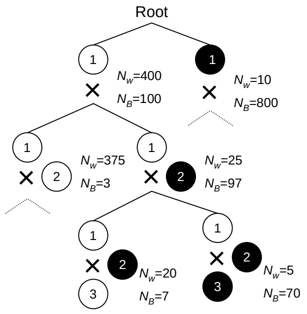

Figure 5. Construction of context tree for a binary image.

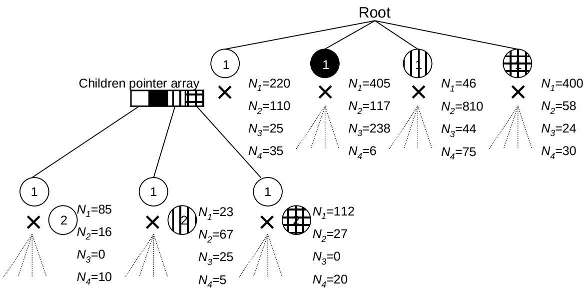

Figure 6. Construction of context tree for a color image.

child pointers and frequency counters as there are colors in the image. The frequency counters (components of the statistics vector) are denoted here as f_1, f_2, … f_k. With a large context size and large number of colors, however, it is unlikely that all colors will appear in a particular node. Our experiments show that for a 25-color image and 15-pixel template, the proportion of non-appearing children pointers and frequency counters can be up to 90% of all memory allocation if linear arrays were used. It is therefore essential to store the children pointers and the frequency vectors as linked lists to optimize memory consumption.

2.3 Pruning the Context Tree

Larger context size allows analyzing of larger structures of the images. However, larger patterns repeat less than smaller patterns and if the size is increased too much, most of the contexts will eventually appear only once or twice. Larger context size tends to make the distribution of the colors in a context more flat. Without enough statistics and clear statistical dominance of one color, the filter is unable to make reliable guess about whether given pixel is noisy, and by which color it could be replaced.

We overcome this drawback by using a pruning technique. Consider a node with the corresponding context c_P, and its children nodes c_1, …, c_k. Denote the number of times the context c_P appears in the image as $N(c_P)$. By definition of CT $N(c_P) = N(c_1) + \ldots + N(c_k)$. When a particular context does not appear frequently enough, it should not be used in filtering. We realize this by applying a simple pruning criterion: if the frequency of a given context falls below a predefined pruning threshold ($\exists\, i$: $N(c_i) \leq$Treshold), the corresponding node is pruned out from the context tree.

By definition of CT, all pixels that appear in a child context c_i appear also in their parent context c_p: $\forall$ $I(x,y) \in c_i$ holds $I(x,y) \in c_P$. When the child context c_i is pruned, traversal in the tree will stop on its parent node, which by definition appears more frequently (or equally frequent in case of only one child) as its child context. The use of pruning criterion guarantees that every context appears in the image frequently enough to be a valid criterion of filtering.

Empirical results support the usefulness of the pruning. Popularity of contexts of size 20 in a sample test image is illustrated in Figure 7. The histogram shows that without pruning most of the contexts (118941) appear only once or twice in the image, and majority of the remaining contexts (21253 + 8971) less frequently than 8 times. Only 6 % of the contexts (about 10000 out of 150000) appear more than 10 times. This means that most of the contexts have too sparse distribution in order to be used for reliable filtering.

Figure 8 illustrates how many pixels are actually filtered in these contexts (filtering with probability threshold 20 %). The less populated contexts (appearing less frequently than 8 times) do not make significant contribution to the filtering. The effect of the pruning is demonstrated in Figure 9 and Figure 10. From Figure 9 one can see that no contexts appearing less than 8 times remain in the tree and Figure 10 shows that the contexts of smaller sizes significantly increase their contribution to the filtering.

3. Noise Models

3.1 Displacement noise

Typically, the map image obtained from a digital scanner is corrupted with specific type of noise. In order to reduce the influence of acquisition device as well as to decrease overall redundancy, that image usually goes through color quantization process. Though pixels of uniform areas are quantized well and are mapped to the same color values, border pixels can be easily mapped to a closer but different color value corrupting the contours of the objects. We refer this kind of noise as *displacement noise*.

We model this type of noise by considering a probability of misplacing the current pixel in a local 3×3 neighborhood. Consider the source image *Source*; the noisy image *Dest* is modeled as follows:

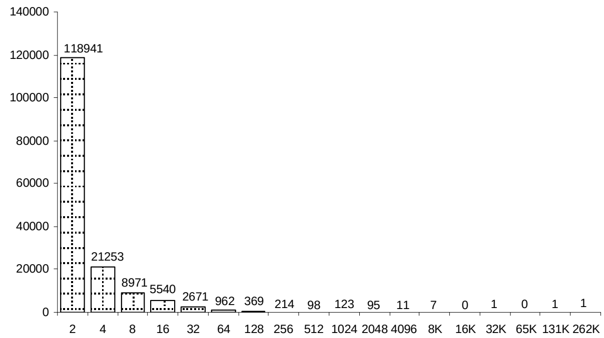

Figure 7. Histogram of contexts popularity for a sample image. The graph represents the number of unique contexts (Y axis) appearing in the image within a given frequency (X axis).

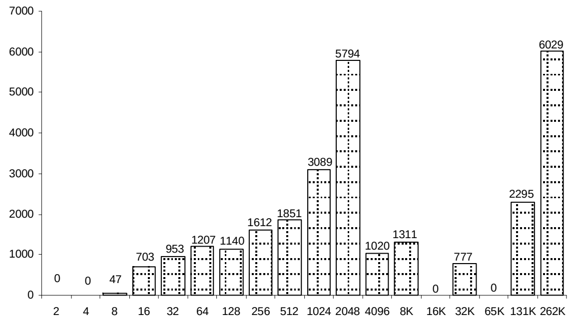

Figure 8. The amount of filtered pixels (Y axis) according to the popularity of the context (X axis) for the statistics of Figure 7.

```
For each Dest(x,y) do
  If rand() < Treshold Then
// do misplacing
    DirX = rand(-1,0,+1)
    DirY = rand(-1,0,+1)
  Dest(x,y)=Source(x+DirX,y+DirY)
  Else
    Dest(x,y) = Source(x,y)
  End If
End For
```

3.2 Impulsive noise

Impulsive noise typically originates from noisy transmitting channels of acquisition devices randomly affecting whole image independently of the region. When the noise level is high, color quantization maps pixels to wrong colors independently of the location of the pixel, and noisy pixels can appear anywhere in the image and can be of any color available in the color palette. We refer this noise as *impulsive noise.* Consider the source image *Source*; the noisy image *Dest* is modeled as follows:

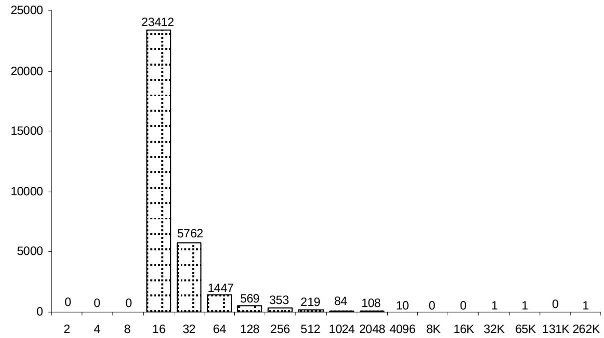

Figure 9. Histogram of the contexts popularity after pruning out contexts appearing less frequently than 8 times.

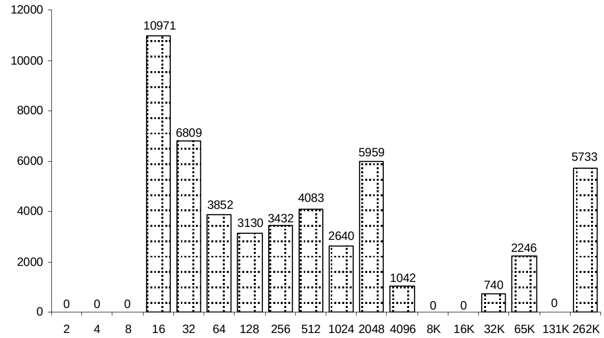

Figure 10. The amount of filtered pixels (Y axis) according to the popularity of the context (X axis) for the statistics of 9.

```
For each Dest(x,y) do
  If rand() < Treshold Then
Dest(x,y)=rnd(1,…,NumberOfColors)
  Else
    Dest(x,y) = Source(x,y)
  End If
End For
```

4. Experiments

We evaluate the proposed Context Tree filter (referred as CT) on a set of six map images chosen from *Finnish National Land Survey* database [15]. Two of them (images #1 and #4) are topographic and the rest are road maps. The images are of different spatial resolution and some of them (images #5 and #6) are affected by quantization noise. In addition to this, we corrupt all images with the noise of two types as described in Section 3.

4.1 Objective evaluation

The proposed filter (CT) is applied with context size 20, probability threshold level 5% and pruning threshold of 128. We compare CT with vector median (VM) [7], adaptive vector median (AVM) [8], morphological (MM) [4] and PGA [9] filters. The efficiency of the filters is evaluated using mean

Table 3. The efficiency of MM, VM, AVM and CT filters measured as Δ*E* distance to the original image for 20% content-dependent (CD) and 5% impulsive noise (I).

	Image 1		Image 2		Image 3		Image 4		Image 5		Image 6	
	CD	I	CD	I	CD	I	CD	I	CD	I	CD	I
MM	23.52	24.37	29.66	30.28	27.75	28.33	14.10	14.48	4.54	8.68	30.45	31.11
VM	3.16	2.51	8.50	7.73	8.58	7.37	3.27	2.46	1.99	1.66	7.81	6.67
AVM	2.51	1.70	4.60	2.46	5.05	3.12	2.18	1.18	1.33	1.15	5.07	3.10
PGA	2.51	1.50	5.48	3.71	5.76	3.79	2.24	1.32	1.75	1.56	5.90	4.02
CT	2.14	0.89	3.95	2.89	3.96	2.44	1.70	0.94	1.19	1.18	3.86	2.94

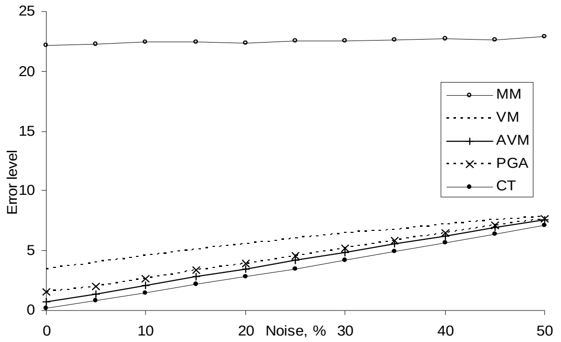

Figure 11. Efficiency of MM, VM, AVM, PGA and CT filters for content-dependent noise.

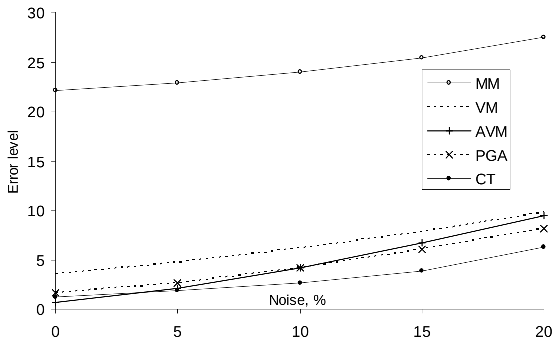

Figure 12. Efficiency of MM, VM, AVM, PGA and CT filters for impulsive noise.

color distance Δ*E* between the original (noiseless) and the filtered images defined by

$$\Delta E = \frac{1}{N}\sum \Delta E^*_{ab}$$

as the normalized sum over all image pixels, where ΔE^*_{ab} is the Euclidean distance between the two color samples in *L*a*b** (CIELAB) uniform color space [16] and is measured as

$$\Delta E^*_{ab} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \; .$$

However, objective distance measure cannot be considered completely relevant for evaluation of the performance because pixelwise measurement does not represent the visual quality. For example when thin and detailed structures are filtered out, this does but it is clearly visible and it corrupts the semantic structures in the map. We therefore present also visual examples of filtered map for subjective evaluation in order to emphasize the ability to preserve repetitive patterns independent of their size.

For content-dependent noise we vary the noise level from 5% to 50% with step of 5%. The results are illustrated in Figure 11. One can see that the proposed filter provides better objective results for all nose levels. On average, the filter outperforms its closest competitor (AVM) by 15%. For impulsive noise we vary the noise level from 5 to 20% with step of 5%; the results are illustrated in Figure 12. The proposed filter outperforms AVM for noise levels higher than 5% noise. On average, CT outperforms AVM up to 30%. Table 3 summarizes the objective measurements for all filters for 20% content-dependent (CD) and for 5% impulsive (I) noise. The measurements are averages over the test set.

4.2 Subjective evaluation

Visual comparisons are presented in Figure 13 for three sample image fragments for 20% content-dependent (CD) and 5% impulsive (I) noise. The VM and AVM filters tend to preserve edges with no blurring. However, thin details of the original data are extensively filtered out since the filters are based on quantitative domination which underlies the median concept. The MM filter is a generalization of gray-scale morphological filter to a color space, and it is based on qualitative dominance. The generalization is considered using *reduced ordering* technique, when an order relation is defined on a vector space by reducing a multivariate object to a single value. For MM filter this order relation is based on a luminance of the color sample [4]. In this way the filter assumes that brighter colors 'dominate' the darker or vice versa. Also, the structuring element defining the operation of the filter is fixed and therefore unable to perform relevant filtering in different areas of the map which have very different structure. All this makes MM filter to perform worst on the selected imagery both by the objective as well as by the subjective comparisons.

The PGA filter performs rather well on impulsive noise. Although some impulses are still visible after one iteration of the filter, they will be removed after

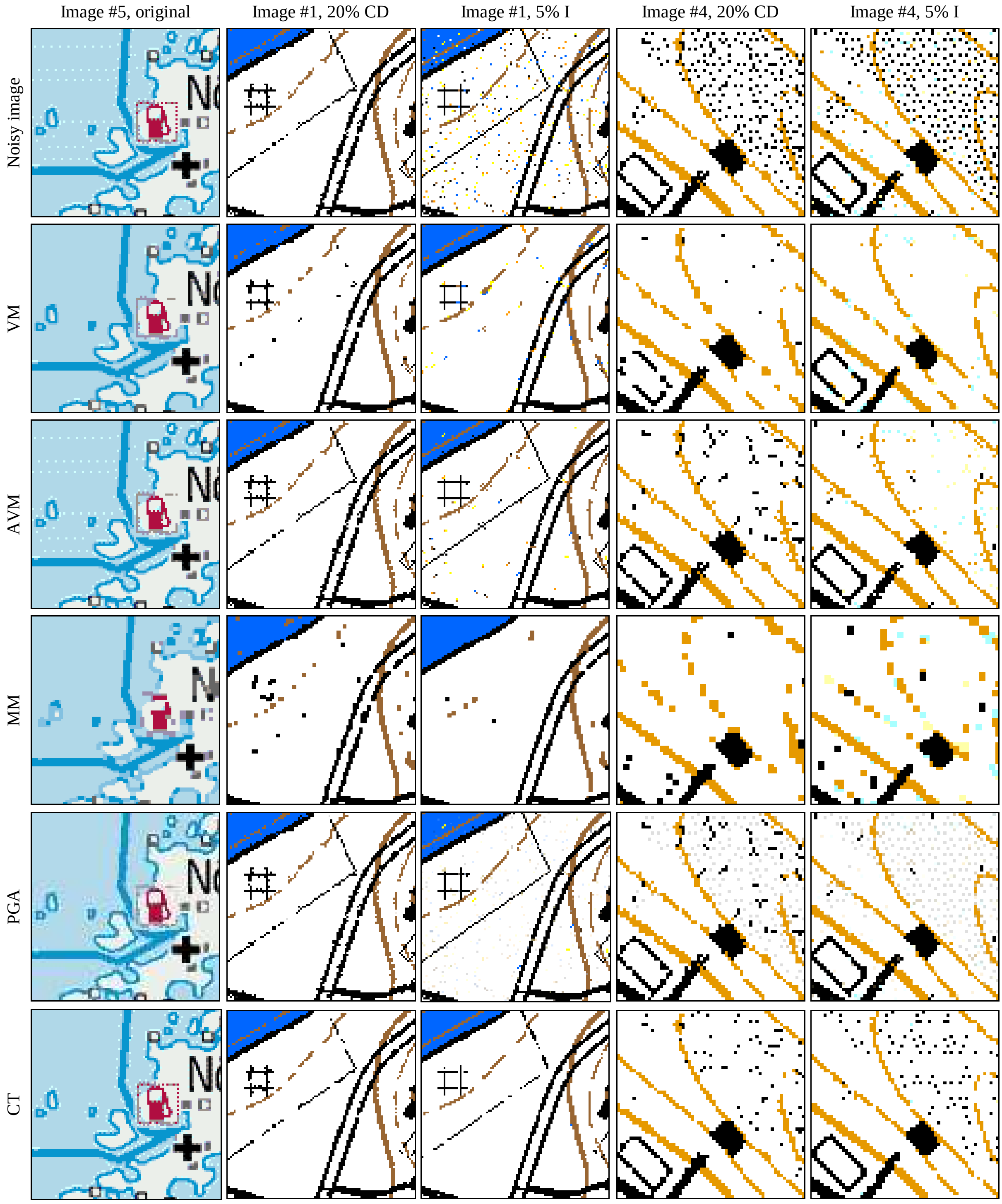

Figure 13. Visual comparison of the competitive filters.

few iterations. However, PGA mostly does not filter the content-dependent noise. This happens because, by its definition, peer group is formed of the neighbor pixels whose color is closest to the processed pixel. In case of content dependent noise, noisy pixels have pixels of the similar (or exactly the same) color in their neighborhood, which makes the peer group averaging ineffective.

In contrary with the competitors, the proposed CT filter deals with statistical domination instead of quantitative or qualitative domination, or distance-based grouping. The filter considers a local pattern to be preserved if it is repeated in the image frequently enough. However, irregular areas (the dotted area in the third example) or patterns not repeated frequently enough are filtered out. This property makes the proposed filter sensitive to the original image data. On the other hand, following the statistical assumptions, the filter is able to restore corrupted structures such as smooth distorted lines and borders. The major condition for the filter to be effective is the statistical consistency of the image; it is therefore mostly suitable for indexed-color palette images and images consisting of computer-generated graphics.

4.3 Processing time

We measure the processing time of the proposed filter (CT) for four selected content-dependent noise levels. The measurement is taken as the average over the test set, and the results are summarized in Table 4. The MM and PGA filters are implemented in Matlab and presented the worst performance, which however originates mostly from the chosen implementation environment. The proposed filer is computationally expensive and its performance depends also on how complicated are the structures in the image.

Table 4. Processing time of the filters under evaluation.

Filter	CT, C++	VM, C++	AVM, C++	MM, Matlab	PGA, Matlab
Time, sec.	15.80	1.34	2.29	20.09	74.14

5. Conclusion

We proposed a statistical filter based on a context tree modeling. The proposed filter is based on a local probability estimation followed by a thresholding replacing less probable patterns with the most probable ones. The filter aims at preserving the repetitive structures of the image, which is an essential property for raster map images. The filter is implemented using a memory efficient management of context tree modeling allowing larger local neighborhood and color depth to be utilized. The size of the context template is dynamically optimized by considering a simple and efficient tree pruning technique.

The performance is compared to vector median, adaptive vector median, color morphological and peer group averaging edge-preserving non-linear filters. The experiments show that the proposed filter outperforms these competitors both in objective and subjective comparisons.

The proposed filter, however, has some limitations of its applicability caused by extensive memory consumption of the algorithm. The filter is considered to be practical for color-indexed palette images when the number of colors is less than or equal to 256, but does not generalize well to true-color images as such. Larger irregular patterns are also not captured very well in case of high noise levels. Nevertheless, the main idea of statistical modeling of repeated structures is more general than relying only statistics within a local neighborhood as done in morphological and peer group filtering.

6. References

[1] Gonzalez, R. C., Woods, R. E., *Digital Image Processing*, 2nd ed., Prentice Hall, 2002.

[2] R. Lukac, B. Smolka, K. Martin, K.N. Plataniotis, and A.N. Venetsanopoulos, "Vector filtering for color imaging", *IEEE Signal Processing Magazine*, vol. 22 (1)74-86, January 2005.

[3] R. Lukac, K.N. Plataniotis, "A taxonomy of color image filtering and enhancement solutions", *Advances in Imaging and Electron Physics*, (ed.) P.W. Hawkes, Elsevier/Academic Press, vol. 140, pp.187-264, June 2006.

[4] M.L. Comer, E. J. Delp, Morphological operations for color image processing, *Journal of Electronic Imaging* 8(3), 279-289, July 1999.

[5] K. N. Plataniotis, D. Androutsos, and A. N. Venetsanopoulos, "Color image filters: The vector directional approach," *Optical Engineering*, vol. 36, no. 9, pp. 2375–2383, 1997.

[6] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: a tutorial," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, issue 3, pp. 157-192, Mar. 1996.

[7] J. Astola, P. Haavisto, Y. Neuvo, "Vector median filters", *Proc. of IEEE*, 78 (4), pp. 678-689, 1990.

[8] R. Lukac, "Adaptive vector median filtering", *Pattern Recognition Letters*, 24, pp. 1889-1899, 2003.

[9] C. Kenney, Y. Deng, B.S. Manjunath, G. Hewer, "Peer group image enhancement", *IEEE Trans. on Image Processing*, vol. 10 (2), 326-334, 2001.

[10] P. Kopylov, P. Fränti, "Filtering of color map images by context tree modeling", *IEEE Int. Conf. on Image Processing (ICIP'04)*, vol. 1, 267-270, October 2004.

[11] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdru, and M. Weinberger, "Universal discrete denoising: Known channel," *IEEE Trans. Inform. Theory*, 51(1):5-28, Jan 2005.

[12] E. Ordentlich, M. Weinberger, and T. Weissman, "Multi-directional context sets with applications to universal denoising and compression," *Proceedings of IEEE Intern. Sympos. Inform. Theory*, 1270-1274, Sep 2005.

[13] J. Rissanen, "A universal data compression system", *IEEE Trans. on Information Theory*, vol. 29 (5), pp. 656–664, 1983.

[14] A. Akimov, A. Kolesnikov and P. Fränti, "Lossless compression of color map images by context tree modeling", *IEEE Trans. on Image Processing*, vol. 16 (1), pp. 114-120, 2007.

[15] National Land Survey of Finland, (http://www.nls.fi).

[16] CIE, Colorimetry, *CIE Pub.* No. 15.2, Centr. Bureau CIE, Vienna, Austria, 1986.