Adaptive Context-Tree-Based Statistical Filtering for Raster Map Image Denoising

Minjie Chen, *Student Member, IEEE*, Mantao Xu, and Pasi Fränti, *Senior Member, IEEE*

***Abstract*—Filtering of raster map images is chosen as a case study of a more general class of palette-indexed images for the denoising problem of images with a discrete number of output colors. Statistical features of local context are analyzed to avoid damage to pixel-level patterns, which is frequently caused by conventional filters. We apply a universal statistical filter using context-tree modeling via a selective context expansion capturing those pixel combinations that are present in the image. The selective context expansion makes it possible to use a much larger spatial neighborhood, with a feasible time and memory complexity, than fixed-size templates. We improve the existing context-tree approaches in two aspects: Firstly, in order to circumvent the context contamination problem, a context-merging strategy is applied where multiple similar contexts are considered in the conditional probability estimation procedure. Secondly, we study a specific continuous-input-finite-output problem in which the map images are corrupted by additive Gaussian noise. Performance comparisons with competitive filters demonstrate that the proposed algorithm provides robust noise filtering performance and good structure preservation in all test cases without any *a priori* information on the statistical properties of the noise.**

***Index Terms*—Context-tree modeling, raster map image, statistical filtering.**

I. Introduction

GEOGRAPHICAL map images are classified into two formats: raster and vector. Vector format is suitable for large databases, providing excellent flexibility for display and compact storage size. A raster image, on the other hand, is encoded in a regular grid of pixel colors in which each color represents a different class of semantic map object. It consists of pixel-level detailed structures and sharp edges but lacks smooth color transitions that are typical for photographic images. It does not require any additional image processing and is suitable for delivery to multimedia applications as such.

Raster maps are an important source of geospatial information due to popularized *geographic information systems*. A large number of topographic maps have been collected during the last century, and they contain geographic features that can be used in the verification of map-image-guided navigation applications. Advances in digital information technology have brought the conversion of these older physical documents into digitized representations. However, the digitization of maps may incur additional noise and errors in the digital versions. Aging of paper archived for a long period also leads to additional random color variations. This kind of image degradation results in mismatch and false recognition of important semantic map objects. Image denoising is therefore needed for accurate conversion of these older maps into raster format. This preprocess can be crucial for the later raster map analysis step, when extracting the semantic content (roads, contours, river) on a map [31]–[34]. Image denoising can also be applied as a preprocessing when converting a raster map into a vector format.

A great variety of noise removal techniques have been investigated for color image processing. However, most noise removal algorithms are developed for only one type of noise model specifically. For instance, to eliminate impulsive noise, a number of denoising algorithms have been developed by first identifying the potential noisy pixels in the color image and then employing a class of vector median filters over those detected noisy pixels. Noisy pixels can be detected either by classifying each pixel directly in RGB color space [1] or by setting some statistical rules in terms of the variation in the local neighborhood [2], [3]. However, these approaches need a training dataset or prior knowledge for constructing the statistical rules. A multi-layer approach was recently proposed [4] to solve the problem in the binary domain.

For additive Gaussian noise, a number of state-of-the-art denoising algorithms have been proposed by selecting an optimal linear combination of a few basis elements in pixel-wise or block-wise order. These pioneer denoising techniques include *wavelet denoising* [5], *non-local means* [6], *dictionary-based method* (K-SVD) [7], *block-matching and 3-D filtering* [8], *Markov random fields*, and *active random fields* [9], [10]. However, they are limited to the case when the true signal can be approximated by a linear combination of a few basis elements, and therefore, they are designed for denoising continuous tone images and do not apply well for palette-indexed images.

Raster map images contain a number of complicated spatial structures such as one-pixel thin lines, textured areas, dashed and dotted lines, text, and symbols. The problem of false filtering exists with most filters designed for photographic imagery when processing these kinds of spatial structures. This is

Manuscript received October 31, 2010; revised March 08, 2011 and June 29, 2011; accepted August 15, 2011. Date of publication August 30, 2011; date of current version November 18, 2011. The work of M. Chen was supported by East Finland Graduate School in Computer Science and Engineering (ECSE), MOPSI project EU (EAKR), Tekniikan edistämissäätiö (TES), and Nokia Scholarship. The work of M. Xu was supported by China NSF (Grant No 61072146) and Shanghai SCST (Grant No 10PJ1404400). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zicheng Liu.

M. Chen and Pasi Fränti are with the School of Computing, University of Eastern Finland, Joensuu, Finland (e-mail: mchen@cs.joensuu.fi; franti@cs.joensuu.fi).

M. Xu is with the School of Electrical Engineering, Shanghai Dianji University, Shanghai, China (e-mail: xumt@sdju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2011.2166538

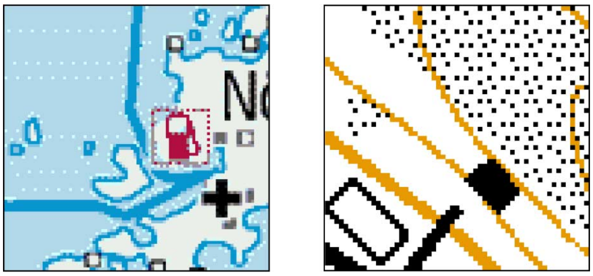


Fig. 1. Examples of complicated structures that are treated as noise by most filters.

because these filters consider local intensity variation as noise but ignore repeated patterns in the entire image. On the other hand, high variance regions including written text, symbols, and textured background lack uniformity but their presence is vital for the readability of the map. Examples of such structures are shown in Fig. 1.

A pioneer work in the art of statistical filtering is the so-called *discrete universal denoising* (DUDE) [11] in design of a filter for binary data with a known noise channel, which comprises two steps: counting statistics for all context patterns encountered (*analysis step*), and denoising by utilizing the conditional probability in a local context (*denoising step*). This method is applicable in denoising of binary images if the error probability δ can be reliably estimated. Namely, if the conditional probability of the current pixel x in its context $P(I(x)\,|\,\mathbf{c})$ is lower than $2\delta(1-\delta)$, it is considered as noise and replaced by its complementary value.

This kind of context-based approach can be extended to the denoising problem with an unknown channel using the min-max criterion [12]. In contrast to the previous algorithms [1]–[10] that incorporate a prior model, statistical filtering is based on an unsupervised learning paradigm. Patterns that are frequently presented in the image are detected and considered as important image structures that should be preserved. On the contrary, pixels that appear seldom in their context are treated as noise and can be filtered out. This allows filtering with preservation of borders and regular structures regardless of their size and variance.

However, the memory allocation for learning the patterns grows exponentially with the number of pixels within the pattern (context), which makes it of limited use in practice. Moreover, the conditional probability estimation becomes inaccurate for those contexts with rare appearance, which is known as the *context dilution problem*. To circumvent this problem, *context-tree modeling* [13], [14], [19] has been applied by pruning redundant nodes of the context tree. This can be done according to different criteria, such as fixed frequency [13], maximum likelihood [15], or the coding cost of the model [14], [16]. Adaptive context selection has also been extended for denoising gray-scale images [17] by using a *minimum description length* (MDL) guided criterion aimed at finding an optimal balance between the variance and bias of the errors in fitting a 2-D *piecewise autoregressive* (PAR) model to input images.

In this paper, we propose a new adaptive context-tree-based statistical filtering algorithm for raster map images. In contrast to the existing context-tree-based methods, we achieve two significant improvements described below.

Firstly, although the context-tree-based algorithms are efficient for the images with a smaller number of noisy pixels, the contexts themselves will embrace a significant number of noisy pixels when noise level is increased in the image. Including noisy pixels (outliers) in the surrounding contexts will make it difficult to estimate a good conditional probability distribution for context models. We call this the *context contamination problem*, when the contexts themselves contain a number of outliers. For those infrequently appearing contaminated contexts, we present a *context-merging* strategy in order to improve conditional probability estimation. In our method, multiple similar contexts are considered jointly in the conditional probability estimation procedure.

Secondly, we extend the algorithm to deal with map images with additive Gaussian noise or mixed Gaussian-impulsive noise. The extended method iteratively conducts a fusion procedure according to the probability distribution of pixels' intensity in the RGB space as well as their conditional probabilities of contexts.

The rest of the paper is organized as follows. The proposed method is introduced in Section II, experimental results are reported in Section III, and finally, conclusions are drawn in Section IV. A preliminary version of this paper has been presented at ICME [30].

II. Proposed Method

For concreteness, a noise-free map image can be formulated as follows: A clean map image $\mathbf{X}$ has M colors (size of color palette) such that the alphabet $\mathrm{A} = \{1, 2, \ldots, M\}$ includes all possible index values in the image. For any pixel $x \in \mathbf{X}$ with index value $I(x)$, its corresponding color in RGB space is

$$\mathbf{X}(x) = \mathbf{m}_{I(x)} = \left[m_{I(x)}(r), m_{I(x)}(g), m_{I(x)}(b)\right] \quad (1)$$

according to the color palette $\mathrm{CP} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_M\}$. Impulsive noise is often produced in the transmission of the clean image signal $I(x)$ over *an M-ary Symmetric Channel* with transition matrix $\mathbf{\Pi}$:

$$\prod(i,j) = \begin{cases} 1-\delta, & i = j \\ \delta/(M-1), & i \neq j \end{cases} \quad (2)$$

where $\Pi(i,j)$ denotes the probability of output index j when the input index is i, and δ is the noise level. The performance of denoising impulsive noise can be evaluated using the following measures:

1) *Error rate*: the percentage of different pixel values between the noise-free and filtered images.
2) *False acceptance rate* (FA): the percentage of noise to survive after filtering (efficiency of noise removal).
3) *False rejection rate* (FR): the probability of an original (clean) image pixel being filtered (amount of corruption imposed by the filter).

Since the clean image $\mathbf{X}$ is also an RGB image, the corresponding noisy image $\mathbf{Y}$ can be produced by adding the same

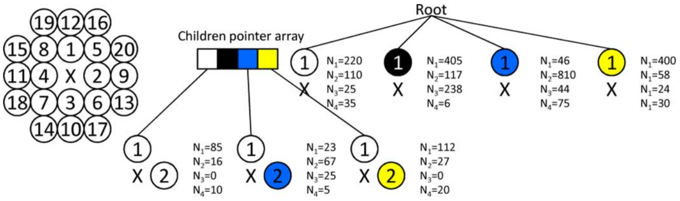

19 12 16
15 8 1 5 20
11 4 X 2 9
18 7 3 6 13
14 10 17
Root
Children pointer array
1 X N1=220 N2=110 N3=25 N4=35
1 X N1=405 N2=117 N3=238 N4=6
1 X N1=46 N2=810 N3=44 N4=75
1 X N1=400 N1=58 N1=24 N1=30
1 X 2 N1=85 N2=16 N3=0 N4=10
1 X 2 N1=23 N2=67 N3=25 N4=5
1 X 2 N1=112 N2=27 N3=0 N4=20

Fig. 2. Part of the context tree (first two levels) and its context template.

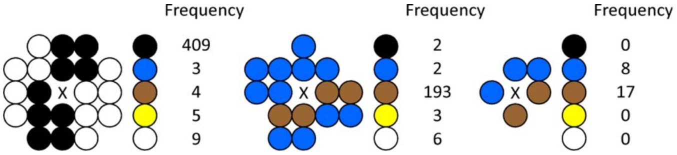


Fig. 3. Examples of context distribution: the pixels in the left and middle contexts are filtered using the dominant color whereas no filtering is done for the pixel in the context on the right.

additive Gaussian noise with variance σ to each color channel independently:

$$\mathbf{Y}(x) = \mathbf{X}(x) + \boldsymbol{\varepsilon}, \text{ where } \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I}) \qquad (3)$$

where $\mathbf{Y}(x)$ and $\mathbf{X}$(x) are the RGB color vectors of pixel x. The quality of denoising of the image with Gaussian noise can be evaluated by:

1) PSNR: Peak signal-to-noise ratio in RGB color space.
2) SSIM: Structural similarity index [21].

A. Context-Based Statistical Filter

Suppose that the clean image $\mathbf{X}$ contains M colors and M is small. A context $\mathbf{c} = \{x_1, \ldots, x_k\}$ can be defined as a set of k pixels. A sample 20-pixel context template is shown in Fig. 2(left), where the current pixel x is marked with "×". For simplicity, a context $\mathbf{c}$ of pixel x is denoted as $x \in \mathbf{c}$. The context $\mathbf{c}$ can be associated with a vector $n_{\mathbf{c}} = (n_{\mathbf{c}}(1), \ldots, n_{\mathbf{c}}(M))$ called a *vector of statistics* for the current pixel x, where $n_{\mathbf{c}}(i)$ is the count statistics of index value i. After the *vectors of statistics* have been collected for every context of the image, the conditional probability of every pixel appearing in its context is estimated as

$$P(I(x) = j \mid x \in \mathbf{c}) = \frac{n_{\mathbf{c}}(j)}{\sum_{i=1}^{M} n_{\mathbf{c}}(i)}. \qquad (4)$$

For simplicity, we denote the conditional probability in the context $\mathbf{c}$ as $P(I(x) \mid \mathbf{c})$.

Statistical filtering can be performed in a two-pass procedure, of which the first pass is the estimation of conditional probabilities (statistical context modeling) and the second pass is the filtering of the noisy pixels. The main idea of the statistical filter follows an assumption that the image signal originates from a universal source. Hence, if the conditional probability $P(I(x) \mid \mathbf{c})$ in context $\mathbf{c}$ is less than a predefined value, the current pixel can be treated as noise and then replaced by the most probable color in the context. Three examples of contexts and their corresponding statistical distributions are demonstrated in Fig. 3. Domination of the most probable color can be observed in the first two examples (left and middle) but not in the last example (right).

B. Context-Tree Modeling

In practical implementation, we optimize memory allocation using *context-tree modeling*. The classical *context-tree modeling* technique has been widely used in the field of data compression with a time complexity of $O(kN)$, where N is the length of the data sequence and k is the depth of the context tree. The tree is built by estimating the count statistics via a sequential traversal of the image pixel by pixel. Each node of the context tree represents a single context by storing the count statistics of each color appearing for the current pixel relative to this context. Since not all possible contexts are present in the image, memory is allocated only for the actual pixel combinations appearing in the image. In our implementation, the spanning of the tree is terminated if the frequency of the context on a given node becomes less than a predefined threshold value T. According to our experiments, there are only 50 000–100 000 contexts for a 20-pixel context template in a 16 color map image, which is far below 16^{20}. An example of context-tree modeling is shown in Fig. 2, where each node represents a single context including the count statistics of each color for the current pixel in respect to the context. An example of the context distribution is shown in Fig. 3.

C. Statistical Filtering by Context-Tree Modeling

Context-tree modeling has been extensively studied in the problem of image compression [28], [29]. In image compression, all pixels must be encoded regardless of the reliability of their contexts. Moreover, one can keep track of the compression performance. Poor probability estimates only lead to a longer code length and thus a large file size. Optimal pruning of a context tree can be done on each node in order to achieve the highest overall compression performance. For instance, a dynamic programming pruning technique was proposed to improve context selection in [14], whereas universal context modeling was employed in [18].

However, conditional probability estimation plays a crucial role in image denoising. Wrongly estimated conditional probability can cause either a lack of detection of a noisy pixel by the algorithm or changing of a clean pixel, causing new noise. In contrast to image compression, several challenges exist when applying statistical context modeling for image denoising. Firstly, the distribution of noise is seldom known, and therefore, it is hard to estimate in practice. Secondly, the contexts themselves may include a significant number of noisy pixels and lead to a so-called *context contamination problem*. If the neighborhood pixels were contaminated by erroneous colors, the particular context would appear infrequently in the image. This causes an inaccurate estimation of the conditional probability distribution. Thirdly, a proper decision rule for the filtering is not trivial to design.

For the improvement of the statistical filter, we will discuss the following three design problems:

a) how to determine the decision rule for filtering;
b) how to calculate the conditional probability of the infrequent contexts;
c) how to estimate the noise level of the image.

a) Decision Rule for Pixel Denoising: Suppose we have a map image with impulsive noise generated by transmitting a clean image $\mathbf{X}$ over *an M-ary Symmetric Channel*. The optimal decision rule in DUDE [11] is essentially a *MAP estimator*, which is

$$u_0 = \arg \max_{I(x)\in\{1,\ldots,M\}} P(I(x)\,|\,\mathbf{c}) \tag{5}$$

where $\mathbf{c}$ is the context of the pixel x. In image filtering, the current pixel x will be replaced by u_0, which is the index value of the highest probability if the decision rule in (6) is met:[1]

$$\begin{aligned} &\frac{(M-1)^2(1-\delta)}{\delta((1-\delta)M-1)}P(I(x)=x_0\,|\,\mathbf{c}) \\ &\quad -\frac{(M-1)}{((1-\delta)M-1)}P(I(x)=u_0\,|\,\mathbf{c})<1, \\ &\qquad\qquad x_0=1,2,\ldots,M. \end{aligned} \tag{6}$$

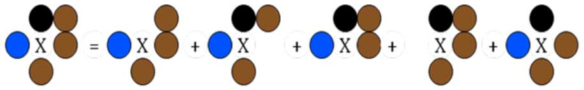
b) Improve Conditional Probability Estimation by a Context-Merging Strategy: Although DUDE follows a so-called "*asymptotic optimality*" property, it requires an infinite sequence of data source for estimating all the conditional distributions of contexts, which is not realistic in practice. In particular, when the context of the pixel is contaminated by erroneous colors, the context can appear infrequently and its associated conditional probability would be far from its true distribution. In order to alleviate this problem, *context-tree modeling* is used by terminating the tree spanning if the frequency of the context becomes less than a given threshold T. In Fig. 5(left), the number of noisy pixels in the context is compared between a fixed-template context and the *context-tree modeling*. No more than two noisy pixels are observed in most contexts regardless of the noise level.

A pruning step is added to remove those contexts as in [13] and [19] if the frequency of a context is less than a predefined threshold T. After pruning, the statistics of its parent node are used for the probability estimation instead. The main challenge for the pruning is that the tree is constructed in a fixed order, and the noisy pixels may appear anywhere in the tree, not just in the leaf nodes. Thus, a clean pixel may also be removed from the context in many cases.

To this end, we will present a *context-merging* strategy for those infrequent contexts that are expected to be contaminated. For each context $\mathbf{c}$, we first construct a set of sub-contexts:

$$S(\mathbf{c}) = \{\mathbf{z}_i\,|\,\mathbf{z}_i = \{\mathbf{c}/x_i\}\}_{i=1}^{k} \tag{7}$$

by removing the ith element from the original context $\mathbf{c}$, where $i=1,\ldots,k$, and $\mathbf{z}_i$ is the sub-context. Without loss of generality, we sum up all the *vectors of statistics* of the sub-contexts as the estimated distribution of the original context $\mathbf{c}$ if the frequency of the context is lower than T. See (8) at the bottom of the page. The idea of this *context-merging* strategy is that the sub-context will appear much more frequently in the image if the noisy pixel is removed from the context. On the contrary, removing a clean pixel will not change the statistics much. For example, in Fig. 4 it is difficult to conclude anything from the original context as it is so infrequent. However, by analyzing the statistics of the sub-contexts we can see that the black pixel is the noisy one and should be removed. After this *context-merging* operation, only noise-free sub-contexts become dominant in the summation of all the sub-context distributions, which serves as a good estimation of the conditional probability for the context model.



Context distribution							Sum
⚫	1	7	3	5	74	4	93
🔵	3	118	4	11	4	30	167
🟤	4	493	5	7	131	25	661
🟡	1	5	1	2	4	1	13
⚪	1	17	1	2	15	2	37

Fig. 4. Example of context-merging strategy. A more reliable context distribution (93, 167, 661, 13, 37) is obtained instead of the estimation (4, 30, 25, 1, 2) obtained by the pruning operation. Colors with low probability (yellow and white) are replaced by the dominant color (brown).

The effect of the *context-merging* strategy is evaluated in Fig. 5(middle and right) when the contexts contain different numbers of noisy pixels. We can observe that the proposed *context-merging* strategy achieves a lower *error rate* and *false acceptance rate* compared to the previous context-pruning method. Even after the *context-merging* process, some sub-contexts are still corrupted by one noisy pixel but they have become frequent enough to be useful. Those contexts can be considered statistically significant for the filtering and they mostly appear in the background region. That is why the proposed *context-merging* strategy retains its efficiency even when two noisy pixels appear in the context.

Performance comparisons are also made for three approaches in Fig. 6(left): a fixed-template context algorithm with different context sizes k, a context-pruning algorithm with different thresholds T, and a *context-merging* algorithm with different thresholds T. We can observe that a lower *false acceptance rate* is achieved at the cost of an increased *false rejection rate* when a greater T is used. In other words, more noisy pixels are filtered out with a trade-off that more clean pixels are replaced by a wrong color. Moreover, it is observed in Fig. 6(middle)

$$P*(I(x)=x_0\,|\,x\in\mathbf{c}) = \begin{cases} P(I(x)=x_0\,|\,x\in S(\mathbf{c})), & \text{if } \displaystyle\sum_{i=1}^{M} n_{\mathbf{c}}(i) < T \\ P(I(x)=x_0\,|\,x\in\mathbf{c}), & \text{otherwise} \end{cases} \tag{8}$$

[1] This decision rule is designed for the count statistics collected on the noisy image. For a clean image, the decision rule is $P(I(x)=x_0\,|\,\mathbf{c}) < \delta/(M-1), x_0 = 1,2,\ldots,M$ instead.

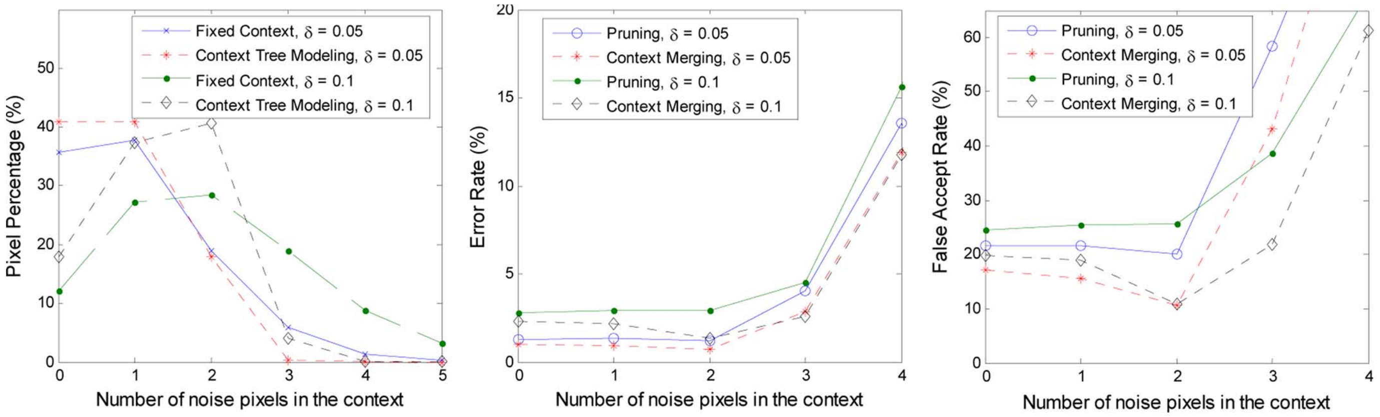


Fig. 5. Comparison of the denoising performance when different numbers of noisy pixels are included in the context. The numbers of noisy pixels in the context are compared for a fixed-template context and context-tree modeling, respectively (left). The *error rate* (middle) and *false acceptance rate* (right) are also evaluated, respectively. Image #1-26 is used is this example with $T = 128$.

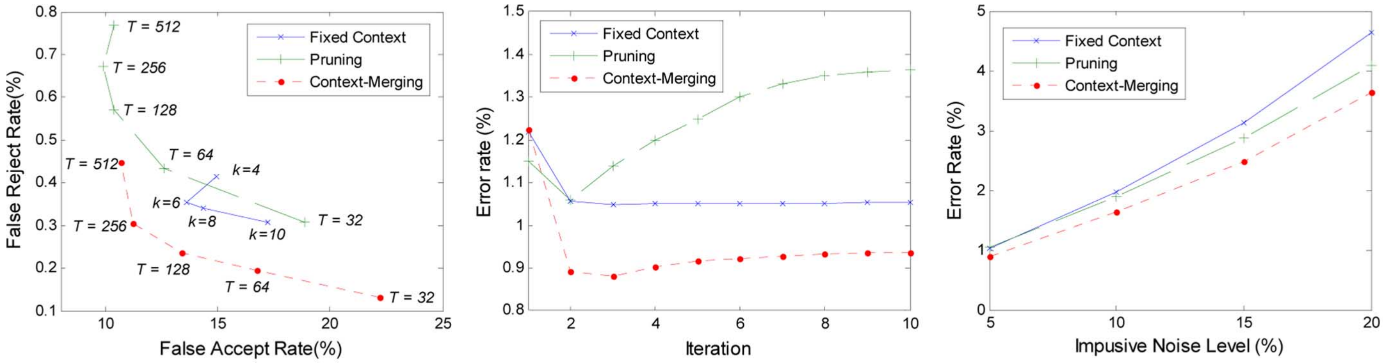


Fig. 6. *False acceptance rate* (FA), *false rejection rate* (FR), and *error rate* are evaluated on different conditional probability estimation algorithms for filtering impulsive noise. The performances are compared by selecting different thresholds T (left), different iterations (middle), and different noise levels (right). Image #2-05 is used as the test image by adding 5% impulsive noise (left and middle), while $T = 128$ and $k = 8$ are used in the middle and on the right.

that further image degradation is not caused even after running several iterations of the proposed statistical filtering algorithm.[2] Image denoising performance is evaluated with different noise levels in Fig. 6(right).

The computational complexity of the merging process is calculated as follows: for each infrequent context, the statistical distributions of M^2 (M is the size of the color palette) similar contexts are identified by tree traversal on the constructed context tree whereas the conditional probability estimation is calculated by summing up all the statistical distributions of the sub-contexts. Suppose that we have an infrequent context $\mathbf{c}$ with k elements; the time complexity of the tree traversal is

$$\sum_{i=1}^{k}((k-i)+(i-1)M) = O(k^2M). \tag{9}$$

c) Noise Level Estimation: In order to improve the filtering robustness under different noise levels, an estimation of noise level δ is needed. It can be estimated either in terms of the min-max criterion [12] or by using image context metrics [22]. However, those solutions conduct the noise estimation in terms of the filtering results for each noise level, which is computationally expensive. A more practical estimate of δ in [23] is the minimized conditional probability in the contexts with "sufficient frequency". In a similar manner, the noise level δ is estimated here on the noisy image directly as

$$\delta = 1 - \max_{\forall y, p(\mathbf{c})>10^{-2}} P(I(y)\,|\,\mathbf{c}) \tag{10}$$

where $P(\mathbf{c})$ is the probability of context $\mathbf{c}$. The performance of the proposed noise level estimation is evaluated on 65 test images, and the results are summarized in Fig. 7. We can observe that the noise level estimation algorithm is reasonably accurate for all tested noise levels.

D. Filtering Additive Gaussian Noise

For completeness, we study the statistical filtering in case of additive Gaussian noise as well. In this case, filtering of raster map images can be considered as a continuous-input-finite-output problem. For a noisy image $\mathbf{Y}$, the problem is defined as finding a denoised palette-indexed image $\mathbf{Z}$ with M colors, which needs an estimation for the color palette. Since the size of the color palette is limited for raster map images, color quantization [20], [24], [25] can be efficiently applied if the color components are well separable [see Fig. 8, (left)]. In the following, several approaches to this problem are addressed.

a) Color Palette Estimation: The nature of the color palette estimation problem is shown in Fig. 8, where five source colors exist in the map image. Some of the colors are

[2] The denoising result of the previous iteration is only used for context pixels. The conditional probability estimation is still based on the noisy input.

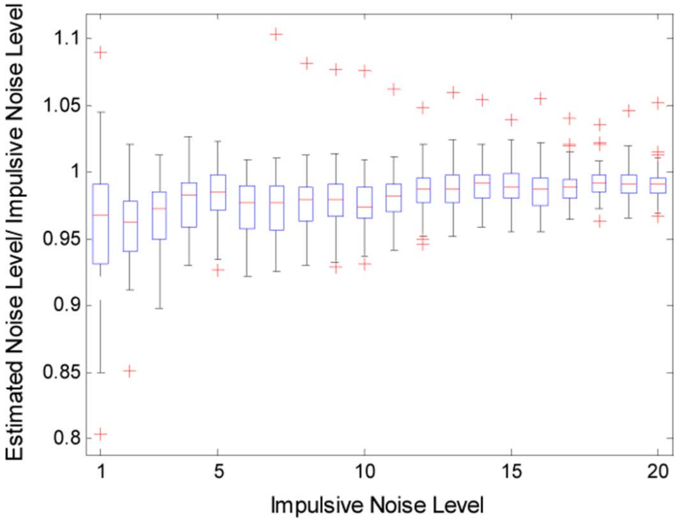


Fig. 7. Performance of noise level estimation on 65 test images.

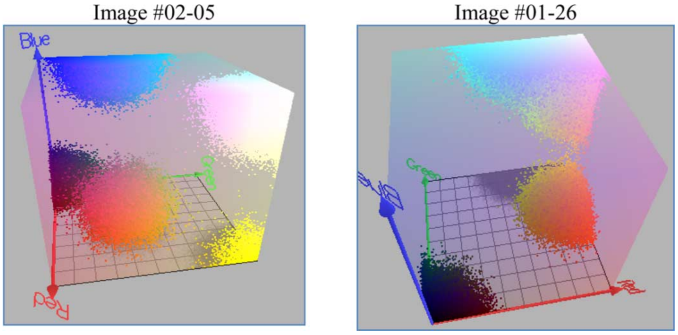


Fig. 8. Distribution in RGB color space for two map images corrupted by Gaussian noise with $\sigma = 25$.

```
Definitions:
    Y: Noisy image
    Z: Denoised image
    I(Z): Index value of image Z
    CP = (m_1, m_2, …, m_M) color palette where
        $\mathbf{m}_i = [m_i(r), m_i(g), m_i(b)], \forall z \in \mathbf{Z}, I(z) \in \{1,..,M\}$ ,
Input: Y
I(Z), CP ← Conduct color quantization according to (13)
σ        ← Estimate the variance according to (16)
For T iterations DO:
    Given I(Z), update P(I(z)|c) according to (8), for $\forall z \in \mathbf{Z}$
    Update I(Z) by fusion procedure according to (14)
    Update CP and σ according to (15) and (16)
End-For
Output: Z, CP.
```

Fig. 9. Pseudo code for filtering Gaussian noise.

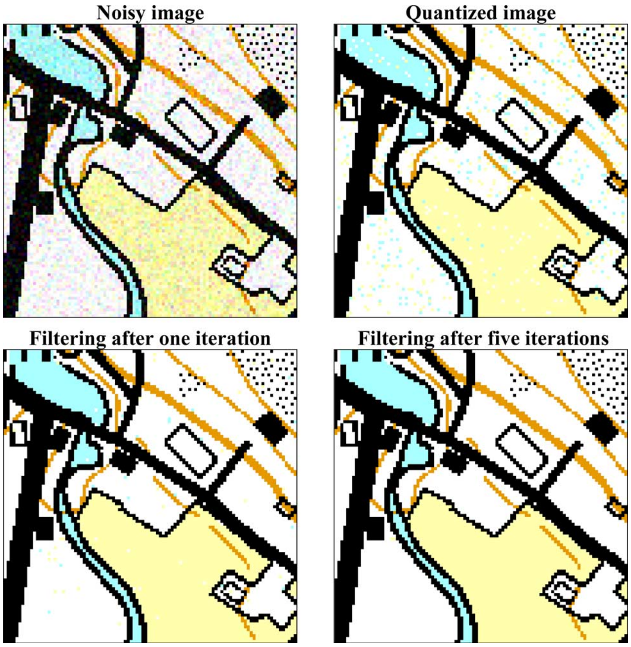

Noisy image
Quantized image
Filtering after one iteration
Filtering after five iterations

Fig. 10. Filtering example (fragment from Image #1-26) of the fusion process for additive Gaussian noise.

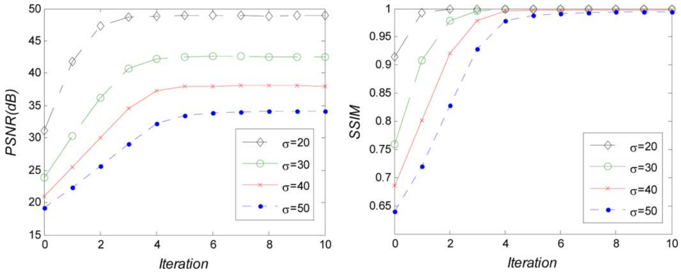


Fig. 11. PSNR and SSIM of the proposed fusion process under different noise levels (#1-26). PSNR and SSIM at iteration 0 are the results of the pre-quantization step.

located near the borders of the RGB color cube. In the noisy images, the colors are spread to form a Gaussian distribution around the source colors. Colors near the border have been truncated, causing the distributions to be one-sided Gaussians. In k-means clustering, this would cause inaccurate estimation of the representative color. We therefore apply a k-medians algorithm where the median value is used (for each color channel separately) to estimate the palette color instead of the mean.

b) Size of the Color Palette M: In color quantization, the size of the color palette can be determined by using a variety of criteria [26] such as the *F-test ratio*, *Bayesian information criterion* (BIC), and *minimum description length* (MDL). For simplicity, we assume that the additive Gaussian noise model shares the same covariance matrix $\sigma^2\mathbf{I}$ ($\mathbf{I}$ is a unit matrix) for each RGB color component and the possible size of the color palette for raster map images is limited to 2–16. This is the operative range for which we expect the algorithm to work well. Accordingly, the size of the color palette is determined as follows:

$$M = \arg\min_{m\in\{2,\ldots,16\}} f(m) \tag{11}$$

$$f(m) = \frac{1}{\sigma^2}\left(\sum_{i=1}^{m} n(i)\sum_{j=1}^{3}(\sigma_{\text{ij}}-\sigma)^2\right)\Bigg/\sum_{i=1}^{m} n(i) \tag{12}$$

where $n(i)$ is the frequency of the colors on each component for the color palette, and σ_{ij} is the variance of the ith component in color channel j.

c) Iterative Fusion Process With Conditional Probability in Context: After color quantization, RGB color space is partitioned into several regions, in which each color vector $\mathbf{Y}(y)$ is represented by its centroid:

$$I(y) = \arg\min_{I(y)\in\{1,\ldots,M\}}\left(\left\|\mathbf{Y}(y)-\mathbf{m}_{I(y)}\right\|^2\right),\quad y\in\mathbf{Y}. \tag{13}$$

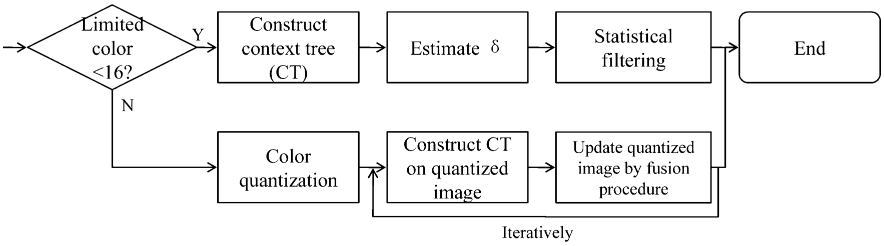

Limited color <16?
Y
N
Construct context tree (CT)
Estimate δ
Statistical filtering
End
Color quantization
Construct CT on quantized image
Update quantized image by fusion procedure
Iteratively

Fig. 12. Workflow of the proposed adaptive context-based method.

Since some color components can overlap (Fig. 8, right), misclassification is inevitable in color quantization (see the quantized image in Fig. 10). To overcome this problem, a novel iterative fusion algorithm is proposed by calculating the distance from a pixel to its color component in the color palette and the conditional probability relative to its context is then applied:

$$I(y) = \arg \min_{I(y)\in\{1,\ldots,M\}} \left(-\log_2 g\left(\mathbf{Y}(y)\,|\mathbf{m}_{I(y)}\right) - \log_2 P(I(y)\,|\,\mathbf{c})\right)$$

$$g\left(\mathbf{Y}(y)\,|\mathbf{m}_{I(y)}\right) = \exp\left(\frac{-\left\|Y(y)-\mathbf{m}_{I(y)}\right\|^2}{2\sigma^2}\right) \qquad (14)$$

where σ is the variance of the additive Gaussian noise and y is the current pixel in the noisy image $\mathbf{Y}$. This fusion filter can be considered as a specific form of the energy function in a *Markov random field* [27], which is derived by replacing the neighborhood similarity with conditional probability in the context.

After the fusion process, the color palette and the estimated noise variance σ are re-estimated as

$$\mathbf{m}_i = \mathrm{median}(\{\mathbf{Y}(y)\,|\,y \in \mathbf{Y}, I(y)=i\}) \qquad (15)$$

$$\sigma = \frac{1}{3|\mathbf{Y}|}\sum_{y\in\mathbf{Y}}\left\|\mathbf{Y}(y)-\mathbf{m}_{I(y)}\right\|^2. \qquad (16)$$

The fusion and the estimation processes are performed iteratively. The pseudo code of the algorithm is described in Fig. 9, and an example of the fusion result is shown in Fig. 10.

Performance comparisons (PSNR and SSIM) are made for different noise levels in Fig. 11. It is observed that the proposed fusion process is effective and very robust for different noise levels. To sum up, the workflow of the proposed adaptive context-tree-based statistical filtering is summarized in Fig. 12.

E. Process Noisy Image With Mixture Noise

To denoise an image with mixed Gaussian-impulsive noise, a straightforward approach is to apply two filters successively: one for impulsive noise and another one for Gaussian noise, respectively. For example, a *fuzzy peer group* [35] combines a statistical method for impulsive noise detection with replacement by an averaging operation to smooth out Gaussian noise.

In a similar manner, the proposed statistical filtering can be extended to the problem of denoising the mixed noise, as outlined in Fig. 13. The proposed extension combines both the case of the statistical filtering for impulsive noise and the case of the fusion process for additive Gaussian noise. Namely, if the DUDE decision rule is met, the current pixel is identified as impulsive noise and then replaced by the color with the maximum conditional probability. Otherwise, the fusion process is applied.

```
Definitions:
    Y: Noisy image
    Z: Denoised image
    I(Z): Index value of image Z
    CP=(m_1, m_2, ..., m_M) color palette where
        $\mathbf{m}_i = [m_i(r), m_i(g), m_i(b)], \forall z \in \mathbf{Z}, I(z) \in \{1,..,M\}$,
Input: Y
I(Z), CP  ← Conduct color quantization according to (13)
$\sigma, \delta$      ← Estimate the noise level according to (10) and (16)
For T iterations DO:
    Given I(Z), update $P(I(z)|\mathbf{c})$ according to (8), $\forall z \in \mathbf{Z}$
    IF DUDE decision rule (6) is met
        Update I(z) by the color with maximum conditional probability
    Else
        Update I(z) by fusion procedure according to (14), $\forall z \in \mathbf{Z}$
    End
    Update CP , $\delta$, and $\sigma$ according to (10), (15), and (16)
End-For
Output: Z, CP.
```

Fig. 13. Pseudo code for filtering mixture noise.

F. Computational Analysis

In general, the context-tree construction leads to a time complexity of $O(kN)$, where k is the size of the context and N is the number of the pixels in the image. Any context can have a maximum of NM/T child nodes, where M is the size of the color palette and T is the frequency threshold for *context merging*. In *context merging*, since the time complexity of every merging process is $O(k^2M)$ in (9), the total time complexity of the *context-merging* process is $O(k^2M \cdot NM/T)$. Additionally, the noise estimation procedure has a time complexity of $O(M)$ in which all contexts with a frequency higher than $p(\mathbf{c}) = 0.01$ are extracted by the tree traversal process. The DUDE decision rule is applied to determine whether or not a pixel is filtered. As the conditional probability of all contexts is pre-calculated, the filtering procedure has a time complexity of $O(N)$. As a result, the total time complexity for denoising impulsive noise is $O(k^2M^2 \cdot N/T)$.

For filtering additive Gaussian noise, the clustering-based color quantization step has a time complexity of $O(MN)$. Context-tree construction and *context merging* have the same complexity as in impulsive noise filtering. In the fusion procedure, the cost function of (14) needs to be calculated on each pixel for all the colors in the color palette, and thus, it leads to a time complexity of $O(MN)$. The total time complexity of denoising additive Gaussian noise is therefore $O(k^2M^2 \cdot N/T)$.

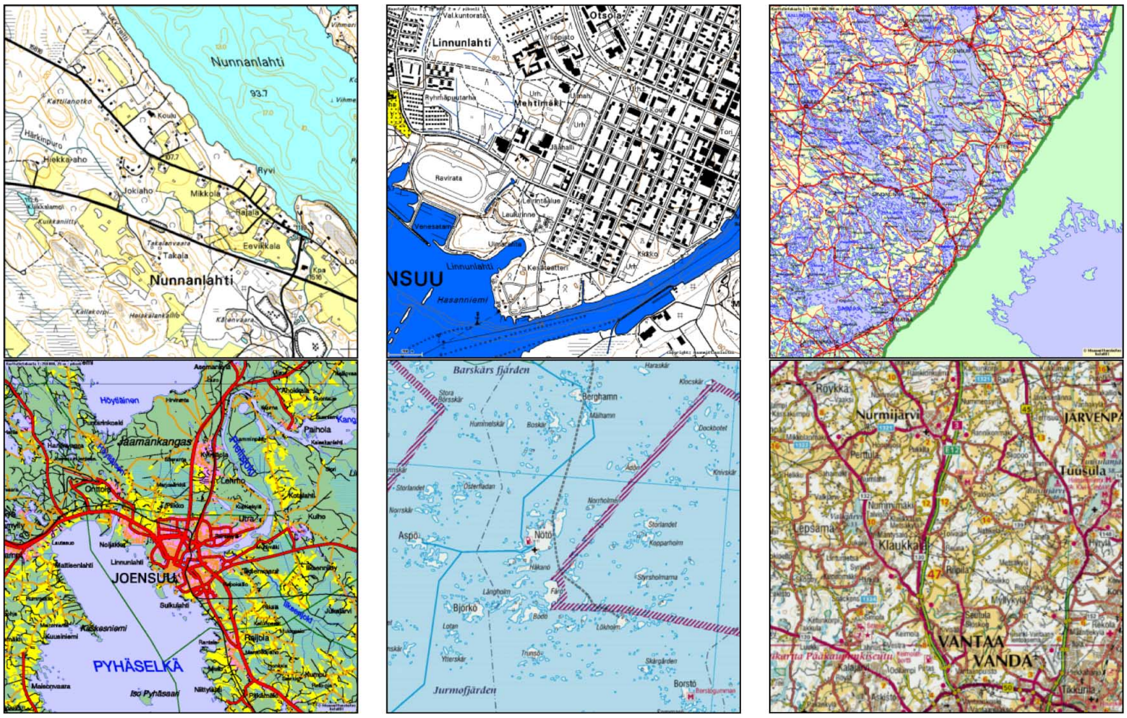


Fig. 14. Sample images from the test set: #1-26, #2-05, #3-03, #4-04, #5-01, and #5-02.

TABLE I
TIME COMPLEXITY OF THE PROPOSED ALGORITHM

Impulsive noise		Additive Gaussian noise	
Step	Complexity	Step	Complexity
Context tree modeling	$O(kN)$	Color quantization	$O(MN)$
Context-merging	$O(k^2M^2{\cdot}N/T)$	Context tree modeling	$O(kN)$
Noise level estimation	$O(M)$	Context-merging	$O(k^2M^2{\cdot}N/T)$
Statistical filtering	$O(N)$	Fusion procedure	$O(MN)$
Total	$O(k^2M^2{\cdot}N/T)$	Total	$O(k^2M^2{\cdot}N/T)$

In case of the mixed noise, either the DUDE decision rule-based statistical filtering or the fusion process is applied, and no additional cost is incurred. The time complexities are summarized in Table I.

III. Experiments

We evaluate the proposed adaptive context-based statistical filtering algorithm (ACS) on a set of map images chosen from the *Finnish National Land Survey* (http://cs.joensuu.fi/sipu/images/mapset.zip); see Table II and Fig. 14. The images have different types (topographic, roadmap) and scales. To test the performance, we artificially distort the images with impulsive noise, with additive Gaussian noise, and with mixed Gaussian-impulsive noise.

A. Parameter Adjustment

In order to choose the appropriate adjustment of the denoising parameters, we analyzed the performance with different selections of parameters. These include the threshold T for the *context merging*, the number of iterations of filtering impulsive noise, and the number of iterations of the fusion procedure.

TABLE II
DESCRIPTIONS OF THE TEST IMAGES

	Scale	Number of images	Image size	Number of colors
Test set #1	1 : 8 000	50	1024 × 1024	5
Test set #2	1 : 20 000	5	1024 × 1024	5
Test set #3	1 : 800 000	4	1024 × 1024	9
Test set #4	1 : 100 000	4	1024 × 1024	10
Test set #5	1 : 250 000	2	800 × 800	16

TABLE III
COMPARISONS OF THE ERROR RATE (TEST SET #1) WHEN CONDITIONAL PROBABILITY ESTIMATION OF A LOCAL CONTEXT IS PERFORMED SEPARATELY OR JOINTLY ON THE WHOLE SET

	Error Rate (%)	
	Single image	Whole test set
$\delta = 0.01$	0.21	0.18
$\delta = 0.05$	0.65	0.58
$\delta = 0.10$	1.31	1.16
$\delta = 0.20$	3.35	3.01

First, we investigate how to select the threshold T. The best selection of T depends on both the number of colors and the noise level of the input image. When the number of colors increases, a larger T should be selected but the exact value of the threshold was found not to be critical for the performance of the algorithm. In our experiments, we set $T = 128$ for test sets #1 and #2 and $T = 256$ for test sets #3–5.

From our experiment, an appropriate selection of the number of iterations in the filtering of impulsive noise can be adopted

TABLE IV
ERROR RATE, FALSE ACCEPTANCE RATE, AND FALSE REJECTION RATE FOR FILTERING IMPULSIVE NOISE

		Error rate (%)					False acceptance rate (%)					False rejection rate (%)				
		AVM	PGF	CT	DUDE	ACS	AVM	PGF	CT	DUDE	ACS	AVM	PGF	CT	DUDE	ACS
δ = 0.05	Set#1	8.65	1.40	1.04	0.94	**0.65**	45.7	26.9	15.6	13.0	7.33	6.70	0.06	0.27	0.30	0.30
	Set#2	4.19	1.26	0.99	0.99	**0.78**	9.89	24.7	14.7	16.5	10.2	3.89	0.02	0.27	0.17	0.28
	Set#3	11.8	3.26	2.41	2.09	**1.59**	47.1	49.3	23.9	36.9	27.5	9.89	0.83	1.28	0.26	0.22
	Set#4	6.55	1.21	1.42	1.34	**0.82**	12.1	20.4	10.5	21.4	11.8	6.25	0.20	0.94	0.28	0.24
	Set#5	9.27	6.02	4.36	2.66	**2.02**	43.6	39.9	25.9	48.3	34.6	7.46	4.23	3.23	0.25	0.30
δ = 0.10	Set#1	10.9	3.56	7.83	1.95	**1.31**	46.3	34.9	77.7	14.5	8.05	6.95	0.07	0.07	0.55	0.56
	Set#2	5.06	3.07	5.01	2.14	**1.57**	11.1	30.5	49.0	18.2	10.7	4.39	0.03	0.13	0.35	0.55
	Set#3	14.1	5.96	4.28	4.14	**3.14**	48.1	52.6	33.1	36.5	26.2	10.3	0.78	1.07	0.55	0.58
	Set#4	7.14	2.45	2.63	2.66	**1.51**	12.9	22.5	17.7	21.4	10.8	6.50	0.22	0.96	0.57	0.48
	Set#5	11.4	7.94	5.89	5.01	**3.85**	44.7	42.0	29.8	44.9	31.5	7.67	4.16	3.23	0.58	0.78
δ = 0.20	Set#1	15.7	10.9	19.9	5.04	**3.35**	48.3	54.3	99.5	21.0	11.7	7.59	0.10	2e-3	1.05	1.28
	Set#2	7.47	9.20	19.5	5.31	**3.49**	14.9	15.8	97.6	23.2	13.3	5.62	0.03	9e-3	0.82	1.05
	Set#3	19.3	12.5	15.6	9.63	**6.84**	51.1	59.8	76.7	44.1	28.4	11.4	0.74	0.31	1.01	1.45
	Set#4	8.65	6.07	12.1	6.67	**3.39**	15.0	29.1	58.8	29.3	13.0	7.05	0.30	0.44	1.02	0.98
	Set#5	16.1	12.8	10.4	10.4	**7.79**	47.5	47.4	41.2	48.0	31.8	8.25	4.09	2.68	1.02	1.80

TABLE V
PSNR AND SSIM FOR FILTERING ADDITIVE GAUSSIAN NOISE

		PSNR					SSIM				
		GSM	NLM	BM3D	ARF	ACS	GSM	NLM	BM3D	ARF	ACS
σ = 15	Set#1	26.8	26.9	29.1	26.2	**59.4**	0.956	0.963	0.987	0.949	**0.999**
	Set#2	27.1	24.9	29.9	25.2	**70.1**	0.944	0.938	0.952	0.939	**0.999**
	Set#3	26.3	23.8	25.6	24.5	**40.8**	0.957	0.946	0.948	0.946	**0.998**
	Set#4	27.1	25.9	27.1	25.2	**51.5**	0.941	0.931	0.945	0.933	**0.999**
	Set#5	26.3	26.0	27.1	26.2	**28.5**	0.881	0.886	0.911	0.872	**0.952**
σ =25	Set#1	24.4	24.9	26.2	24.2	**47.1**	0.943	0.944	0.978	0.942	**0.999**
	Set#2	24.0	23.5	26.5	23.3	**59.2**	0.919	0.918	0.941	0.924	**0.999**
	Set#3	23.7	22.4	23.7	23.0	**27.0**	0.919	0.904	0.928	0.929	**0.975**
	Set#4	24.6	24.1	25.5	23.8	**33.4**	0.874	0.881	0.923	0.902	**0.988**
	Set#5	25.1	24.3	26.1	25.1	**27.7**	0.846	0.827	0.899	0.860	**0.926**
σ = 35	Set#1	22.2	22.7	23.9	22.3	**42.7**	0.898	0.912	0.964	0.919	**0.998**
	Set#2	21.6	21.6	23.9	21.5	**48.4**	0.869	0.893	0.926	0.893	**0.999**
	Set#3	21.1	20.7	21.6	21.3	**25.7**	0.823	0.857	0.859	0.860	**0.968**
	Set#4	21.5	22.0	22.9	21.8	**31.3**	0.717	0.821	0.783	0.771	**0.979**
	Set#5	22.1	22.6	23.9	22.0	**25.4**	0.700	0.766	0.796	0.766	**0.912**

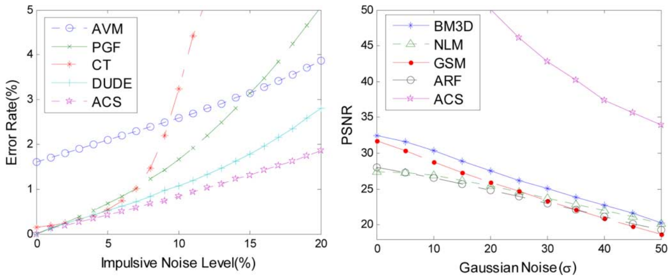


Fig. 15. Performance comparison with different noise levels (Image #01-26 is used).

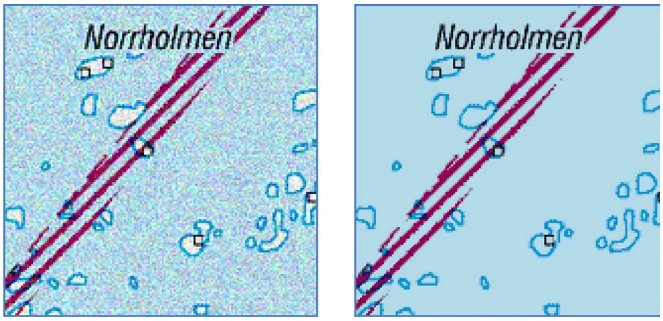


Fig. 16. Worst case example of the denoising result for image #05-01: noisy image (left) and the denoised image (right). Noise has been eliminated; text and other details are preserved. However, due to the sub-optimal color palette estimation, the light blue and white colors have been merged, which leads to the loss of essential information.

as follows: two iterations are conducted for images of impulsive noise with noise levels smaller than 10% and three iterations are conducted for images with noise levels greater than 10%. For denoising additive Gaussian noise, five iterations provide a good compromise between denoising performance and computational complexity (see Fig. 11). For denoising mixed Gaussian-impulsive noise, five iterations of the fusion process are conducted for Gaussian noise denoising, and two iterations

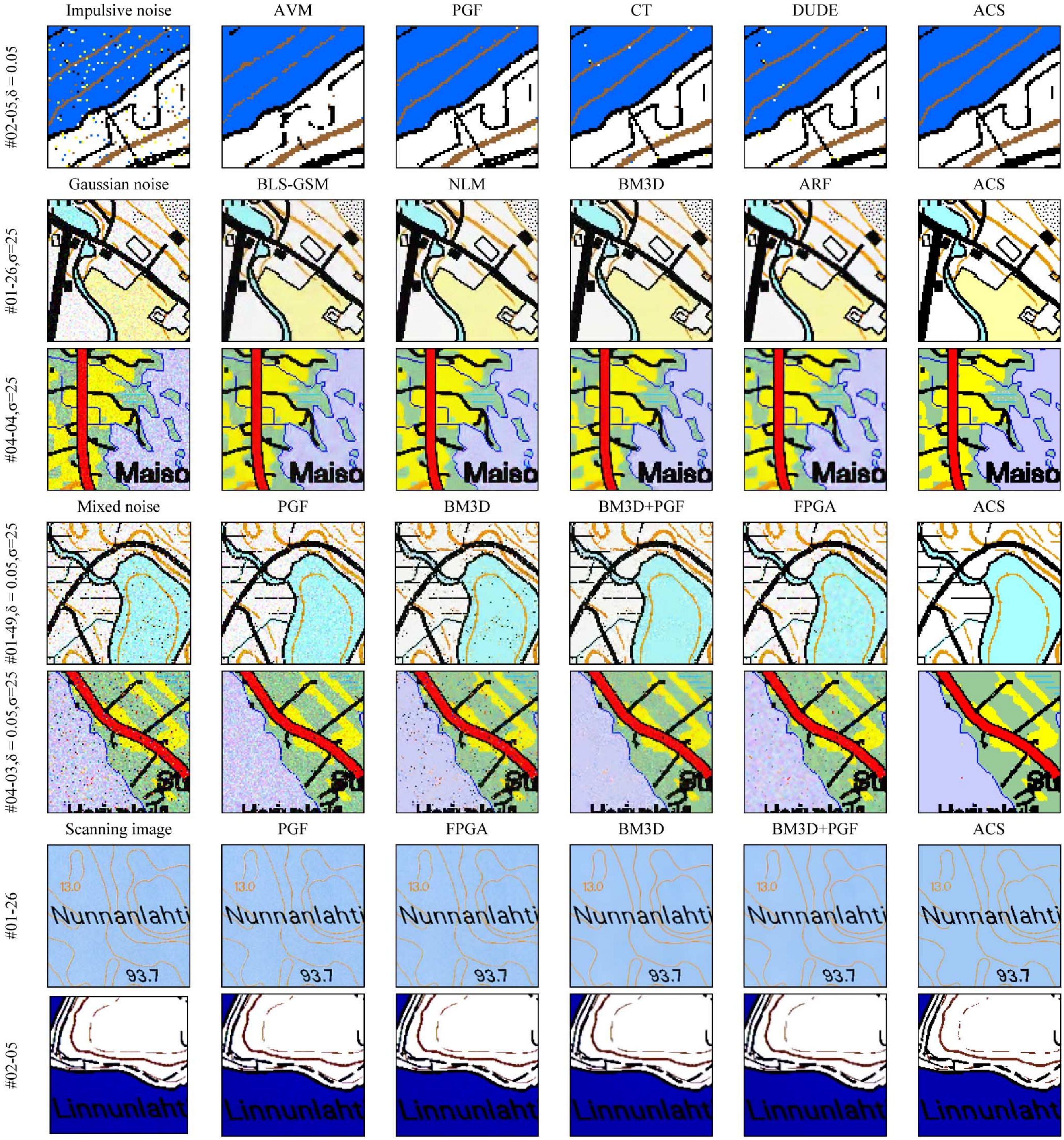


Fig. 17. Visual performance comparison.

of DUDE decision rule-based statistical filtering are performed for detecting and denoising impulsive noise.

Test set #1 consists of 50 topographic images of size 1024 × 1024, which all use the same types of notations to represent topographic information and include the same patterns. We therefore test whether the denoising performance will improve when the conditional probability estimation is estimated on the entire test set using the same conditional probability of each context for all the images. From our experiments, the algorithm performance improves by 10% compared with the case of conditional probability estimation on a single image (see Table III). To sum up, if the type of image is known, the context modeling can be trained and better performance is achieved. Nevertheless, this approach is not used in the following experiments.

B. Objective Evaluation

a) Impulsive Noise: First, we compare the proposed algorithm with four alternative filters: *adaptive vector median* (AVM) [2], *fast peer group filter* (PGF) [3], *context-tree filter*

TABLE VI
PSNR AND SSIM FOR FILTERING MIXED NOISE

		PSNR					SSIM				
		PGF	BM3D	BM3D +PGF	FPGA	ACS	PGF	BM3D	BM3D +PGF	FPGA	ACS
$\delta = 0.03$, $\sigma = 15$	Set#1	21.4	20.2	22.5	22.0	**25.6**	0.769	0.786	0.940	0.901	**0.977**
	Set#2	21.3	19.8	2201	22.9	**23.7**	0.772	0.759	0.906	0.891	**0.978**
	Set#3	21.4	20.9	22.0	23.0	**25.4**	0.779	0.798	0.902	0.906	**0.970**
	Set#4	22.7	21.9	23.8	25.4	**25.8**	0.710	0.800	0.898	0.860	**0.967**
	Set#5	21.4	22.3	23.7	24.8	**25.9**	0.652	0.739	0.868	0.821	**0.945**
$\delta = 0.05$, $\sigma = 25$	Set#1	18.0	17.7	19.8	19.5	**23.6**	0.635	0.685	0.867	0.788	**0.966**
	Set#2	18.1	17.2	19.4	19.5	**21.7**	0.669	0.670	0.830	0.750	**0.967**
	Set#3	17.9	18.5	19.7	20.7	**23.0**	0.654	0.707	0.817	0.785	**0.952**
	Set#4	19.0	19.1	21.3	21.4	**24.4**	0.598	0.666	0.780	0.716	**0.961**
	Set#5	18.3	19.9	21.9	21.5	**22.7**	0.523	0.640	0.774	0.673	**0.914**
$\delta = 0.10$, $\sigma = 35$	Set#1	14.8	14.3	16.3	16.4	**20.3**	0.499	0.514	0.649	0.649	**0.925**
	Set#2	15.0	13.7	16.1	16.1	**18.7**	0.572	0.532	0.654	0.637	**0.929**
	Set#3	15.3	15.3	16.6	17.6	**19.6**	0.544	0.558	0.625	0.672	**0.895**
	Set#4	16.4	15.2	17.6	18.1	**21.5**	0.505	0.479	0.564	0.594	**0.920**
	Set#5	16.2	16.3	18.4	18.5	**19.8**	0.425	0.474	0.547	0.551	**0.895**

(CT) [13], [19], and *discrete universal denoising* (DUDE) [11], [14] by using images corrupted by impulsive noise. The performance is measured by the *error rate* (%), *false acceptance rate* (%), and *false rejection rate* (%) (see Table IV). Experiments show that the algorithms based on statistical filtering (CT, DUDE, ACS) achieve better noise reduction performance (lower *false acceptance rate*) and better preservation of image details than the best of the conventional filters (AVM, PGF). This is because raster map images consist of pixel-level detailed structures, sharp edges, and repetitive patterns whereas those traditional algorithms are based on *a priori* assumption that the images have smooth color transitions.

The context-tree filter with a predefined filtering threshold lacks robustness in denoising images with different noise levels. However, both the noise estimation and DUDE decision rule have been exploited here to achieve robust performance in the proposed statistical filter. Since the conditional probability estimation is further improved by the *context-merging* strategy, the proposed algorithm achieves better performance than DUDE, which has been the best solution for universal discrete denoising problems so far.

b) Additive Gaussian Noise: Four state-of-the-art filters for denoising Gaussian noise are also evaluated: *wavelet denoising using Gaussian scale mixtures* (BLS-GSM) [5], *non-local means* (NLM) [6], *block matching and 3-D filtering* (BM3D) [8], and *active random fields* (ARF) [10]. Their performance comparisons are summarized in Table V using the *peak signal-to-noise ratio* (PSNR) and *structural similarity index* (SSIM) and are visually demonstrated in Fig. 17. It can be observed that the proposed method achieves both visually and numerically better results than the comparative filters. The difference is remarkable for test sets #1–4. This is because the comparative algorithms are designed with *a priori* assumption of smooth color transitions in the images. Moreover, raster map images have a very limited output and this is not considered in those algorithms.

We also compared the performances between the images with different noise levels as shown in Fig. 15. It can be observed that the proposed algorithm is also robust to the noise level. However, the performance is less impressive for image #05-01 than for the other images. This is because most color components become more overlapped and those color components lack separation in color space when its number increases. In the color palette estimation of Section II-D, those overlapped color components are merged into the same output color. This problem is demonstrated in Fig. 16 for #05-01, in which the number of output colors is reduced to nine, and it degrades the denoising performance.

c) Mixed Noise: We compare the performance using images with mixed Gaussian-impulsive noise against PGF,[3] BM3D, and *Fuzzy Peer Group Averaging* (FPGA). An additional experiment was carried out by using BM3D as a first step for denoising Gaussian noise followed by PGF as a second step for denoising impulsive noise. The results are reported in Table VI and Fig. 17. Both the numerical results and visual examinations show that the proposed algorithm is superior to all the comparative algorithms in denoising images with mixed Gaussian-impulsive noise.

d) Real-World Examples: We performed additional tests by printing and re-scanning two selected images from set #1 and #2; see Fig. 17 for scanning image. The resulting images have slightly different colors than their original ones and Gaussian type of noise and blurred contours also appear. Among the other filters, PGF cannot remove the Gaussian-type of noise in these real examples, and BM3D causes blurring effect around the contours. The proposed method (ACS) preserves the larger structures in Set #2 very well but some of the noisy thin structures are broken. For the image from set #1, thin structures are mostly well restored. No false colors or blurring effects appear in the output image either, but discontinuation of thin contours appears at places.

[3] DUDE and the CT algorithm can only be used for discrete denoising problems and cannot be applied when the input is a continuous-tone image.

IV. Conclusion

We have proposed a statistical filtering algorithm dealing with map images distorted by impulsive noise, additive Gaussian noise, and mixed Gaussian-impulsive noise. The proposed filter incorporates an information fusion process which exploits both the color distribution in RGB space and the conditional probabilities of a given pixel in a local context. It operates with no prior knowledge of the properties of the noise and aims at maximal preservation of repetitive structures of the image. This is an essential property for raster map images and is expected to generalize to other types of palette-indexed imagery as well. It can also be viewed as a pioneer study to attack distortion caused by unknown noise types. Experiments with different noise types and spatial image resolutions show that the proposed filter provides robust and reliable filtering performance and good structure preservation ability.

We also investigate the *context contamination* problem in conditional probability estimation in statistical filtering and a *context-merging* strategy is proposed to improve the estimation accuracy for those infrequent contexts.

The proposed algorithm can also be used for other types of color palette images such as engineering drawings, schemes, comic books, and similar art imagery. Raster map images have the typical properties (sharp edges and repeated patterns) that also exist in other kind of images, and were therefore selected here as a typical but challenging case study for evaluating the efficiency of the proposed algorithm.

Future work can be done to extend the proposed filtering method to continuous-tone images in addition to color indexed images. More theoretical analysis is needed on how to select the optimal threshold in context merging.

Acknowledgment

The authors would like to thank the reviewers and the editor for their valuable comments and suggestions, which have been very useful in improving the technical content and the presentation of the paper. The authors also would like to thank Prof. S. Morillas for providing the code of his algorithm.

References

[1] H. Zhou and K. Mao, “An impulsive noise color image filter using learning-based color morphological operation,” *Digit. Signal Process.*, vol. 18, pp. 406–421, 2008.

[2] R. Lukac, “Adaptive vector median filtering,” *Pattern Recognit. Lett.*, vol. 24, pp. 1889–1899, 2003.

[3] B. Smolka and A. Chydzinski, “Fast detection and impulsive noise removal in color images,” *Real-Time Imaging*, vol. 11, no. 5–6, pp. 389–402, 2005.

[4] M. Chen, M. Xu, and P. Fränti, “Multi-layer filtering approach for map images,” in *Proc. IEEE Int. Conf. Image Processing*, 2009, pp. 3953–3956.

[5] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli, “Image denoising using a scale mixture of Gaussians in the wavelet domain,” *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.

[6] A. Buades, B. Coll, and J. M. Morel, “A non local algorithm for image denoising,” in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2005, vol. 2, pp. 60–65.

[7] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[9] S. Roth and M. J. Black, “Fields of experts: A framework for learning image priors,” in *Proc. IEEE Int. Conf. Computer Vision*, 2005, pp. 860–867.

[10] A. Barbu, “Training an active random field for real-time image denoising,” *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2451–2462, Nov. 2009.

[11] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. Weinberger, “Universal discrete denoising: Known channel,” *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 5–28, Jan. 2005.

[12] G. Gemelos, S. Sigurjonsson, and T. Weissman, “Algorithms for discrete denoising under channel uncertainty,” *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2263–2276, Jun. 2006.

[13] P. Kopylov and P. Fränti, “Filtering of color map images by context tree modeling,” in *Proc. IEEE Int. Conf. Image Processing (ICIP’04)*, 2004, vol. 1, pp. 267–270.

[14] E. Ordentlich, M. J. Weinberger, and T. Weissman, “Multi-directional context sets with applications to universal denoising and compression,” in *Proc. 2005 IEEE Int. Symp. Inform. Theory (ISIT’05)*, 2005, pp. 1270–1274.

[15] G. Gimel’farb, “Adaptive context for a discrete universal denoiser,” in *Proc. Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR Int. Workshops, SSPR 2004 and SPR*, 2004, pp. 477–485.

[16] A. Akimov, A. Kolesnikov, and P. Fränti, “Lossless compression of color map images by context tree modeling,” *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 114–120, Jan. 2007.

[17] G. Zhai, X. Wu, X. Yang, and W. Zhang, “MDL context modeling of images with application to denoising,” in *Proc. IEEE Int. Conf. Image Processing*, 2009, pp. 3845–3848.

[18] J. Yu and S. Verdú, “Schemes for bidirectional modeling of discrete stationary sources,” *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4789–4807, Nov. 2006.

[19] A. Podlasov, P. Kopylov, and P. Fränti, “Statistical filtering of raster map images using a context tree model,” in *Proc. Int. Conf. Signal-Image Technology and Internet-Based Systems (IEEE-SITIS)*, 2007, pp. 467–474.

[20] Q. Zhao, V. Hautamäki, I. Kärkkäinen, and P. Fränti, “Random swap EM algorithm for finite mixtures models in image segmentation,” in *Proc. IEEE Int. Conf. Image Processing*, 2009, pp. 2397–2400.

[21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[22] X. Zhu and P. Milanfar, “A no-reference image content metric and its application to denoising,” in *Proc. IEEE Int. Conf. Image Processing*, 2010, pp. 1145–1148.

[23] E. Ordentlich, G. Seroussi, S. Verdú, M. Weinberger, and T. Weissman, “A discrete universal denoiser and its application to binary images,” in *Proc. IEEE Int. Conf. Image Processing*, 2003, pp. 117–120.

[24] P. Kopylov and P. Fränti, “Color quantization of map images,” in *Proc. IASTED Conf. Visualization, Imaging, and Image Processing (VIIP’04)*, 2004, pp. 837–842.

[25] D. Mavridis and N. Papamarkos, “Color quantization using principal components for initialization of Kohonen SOFM,” in *Proc. IEEE Int. Conf. Image Processing*, 2009, pp. 1633–1636.

[26] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “Cluster validity methods,” in *ACM SIGMOD Record*, 2002.

[27] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, “A comparative study of energy minimization methods for Markov random fields with smoothness-based priors,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 1068–1080, Jun. 2008.

[28] J. Rissanen, “A universal data compression system,” *IEEE Trans. Inf. Theory*, vol. IT-29, no. 5, pp. 656–664, Sep. 1983.

[29] M. Weinberger, J. Rissanen, and R. Arps, “Application of universal context modeling to lossless compression of gray-scale images,” *IEEE Trans. Image Process.*, vol. 5, no. 4, pp. 575–586, Apr. 1996.

[30] M. Chen, M. Xu, and P. Fränti, “Statistical filtering of raster map images,” in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME’10)*, 2010, pp. 394–399.

[31] S. Leyk and R. Boesch, “Colors of the past: Color image segmentation in historical topographic maps based on homogeneity,” *Geoinfomatica*, vol. 14, pp. 1–21, 2010.

[32] A. Khotanzad and E. Zink, "Contour line and geographic feature extraction from USGS color topographical paper maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 1, pp. 18–31, Jan. 2003.

[33] Y. Chen, Y. Liu, Z. Fu, and R. Wang, "Automatic extracting residential areas from color scanned topographical maps," *Image Signal Process.*, 2009.

[34] T. Henderson and T. Linton, "Raster map image analysis," in *Proc. Int. Conf. Document Analysis and Recognition*, 2009, pp. 376–380.

[35] S. Morillas, V. Gregori, and A. Hervas, "Fuzzy peer groups for reduction mixed Gaussian-impulsive noise from color images," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1452–1466, Jul. 2009.

Minjie Chen (S'10) received the B.Sc. and M.Sc. degrees in biomedical engineering from Shanghai Jiaotong University, Shanghai, China, in 2003 and 2007, respectively. Since 2008, he has been pursuing the Ph.D. degree in computer science at the University of Eastern Finland, Joensuu, Finland.

His research interests include image denoising and compression, spatial-temporal data compression, and medial image analysis.

Mantao Xu received the B.Sc. degree in mathematics from Nankai University, Tianjin, China, in 1991, the M.Sc. degree in applied mathematics from Harbin Institute of Technology, Harbin, China, in 1997, and the Ph.D. degree in computer science from the University of Joensuu, Joensuu, Finland, in 2005.

He served as a Research Lab Manager with Kodak Health Group and Carestream Health Inc., Global R&D Center, Shanghai, China, from 2005 to 2010. He is now a Research Professor at the School of Electric Engineering, Shanghai Dianji University, Shanghai, China. His research interests include medical image analysis, multimedia technology, and pattern recognition.

Pasi Fränti (SM'08) received the M.Sc. and Ph.D. degrees in science from the University of Turku, Turku, Finland, in 1991 and 1994, respectively.

Since 2000, he has been a Professor of computer science at the University of Eastern Finland, Joensuu, Finland. He has published 56 journal and 128 peer review conference papers, including 10 IEEE transaction papers. His research interests include clustering algorithms, vector quantization, lossless image compression, voice biometrics, and location-based systems. He has supervised 14 Ph.D. students and is currently the head of the East Finland doctoral program in Computer Science and Engineering (ECSE).

Dr. Fränti serves as an Associate Editor for *Pattern Recognition Letters*.