Adaptive Filtering of Raster Map Images Using Optimal Context Selection

Minjie Chen[1], *Mantao Xu*[2], *Pasi Fränti*[1]
[1]University of Eastern Finland, Finland [2]Shanghai Dianji University, China

ABSTRACT

Filtering of raster map images or more general class of palette-indexed images is considered as a discrete denoising problem with finite color output. Statistical features of local context are used to avoid damages of some specific but frequently occurring contexts caused by conventional filters. Several context-based approaches have been developed using either fixed context templates or context tree modeling. However, these algorithms fail to reveal the local geometrical structures when the underlying contexts are also contaminated. To address this problem, we propose a novel context-based voting method to identify the possible noisy pixels, which are excluded in the context selection and optimization. Experimental results show that the proposed context based filtering outperforms all other existing filters both for impulsive and Gaussian additive noise.

Index Terms *—Nonlinear filters, context modeling*

1. INTRODUCTION

Raster map images are commonly encoded in a regular grid of pixel colors arrayed in rows and columns, 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 procedure and is therefore suitable for the delivery to the multimedia applications. However, such images can be degraded in image acquisition (digitization) process and this color degradation may lead to severe false recognition of important semantic map objects. Hence, image filtering is needed before publishing the image. There are several technical challenges for designing suitable filter for this kind of images.

A great variety of noise removal techniques have been extensively investigated for color image processing. The multi-layer approach in [1] converts the problem into binary domain whereas other approaches try to work in the color domain. However, these algorithms are usually developed in terms of specifically noise model, such as *impulsive noise* [2-3] and *additive Gaussian noise* [4-7]. In the case of *impulsive noise*, noisy pixels could be detected if suitable statistical rules based on the local variation were designed. However, these rules are always based on presumable knowledge, which inevitably incur a significant misclassification error. For *additive Gaussian noise,* most of filters are designed by selecting an optimal linear combination of a few basis elements, either pixel-wise or block-wise. These methods assume that the true signal can be approximated by a linear combination of few basis elements and designed only for continuous tone images. They are not applicable for map images in general, because raster map images need a finite color output. Moreover, the repeatable geometrical structures inherited in map images haven't been considered in these methods. In other words, the literature lacks of methods that is capable to filter this kind of images properly.

A pioneer work in the art of statistical filtering is the so-called *discrete universal denoising* (DUDE) [8] for filtering binary data with a known noisy channel. It consists of two steps: counting statistics for all context patterns encountered, and denoising by utilizing the conditional probability of local context. This method is applicable in denoising of binary image if the error probability δ can be estimated presumably. Namely, for a given pixel x, if the conditional probability in the surrounding context $P(x|\mathbf{c})$ is lower than $2\delta(1-\delta)$, it would be treated as noise pixel and replaced by the complementary value. However, the memory allocation for running this algorithm grows exponentially with the size of the fixed context, which makes the implementation of this algorithm intractable. To circumvent this memory allocation problem, the context tree modeling [9, 10] has been applied by pruning redundant nodes of contexts.

In practice, albeit the above algorithms are very efficient for the input images with a few number of noise pixels, the contexts themselves will embrace a significant number of noise pixels when noise level is increased for the input image. Even refining the contexts adaptively using the pruning algorithm [10-12] was not able to remedy this open problem completely. Obviously, including noise pixels or outliers in the surrounding contexts will make it extremely difficult to estimate a good conditional probability distribution for context modeling. Thus, the underlying noise pixels in the selected contexts must be first detected and then need to be excluded in the modeling.

The motivation of this work is to attack the problem by removing detected outliers from the context by classifying the context according to its filtering efficiency. Each problematic context is processed by removing one pixel in turn. In case of significant change in the self-entropy of the context, we conclude that the removed pixel is noisy one.

The work was supported by MOPSI project EU (EAKR), Natural Science Foundation of China (Grant No 61072146), Shanghai Committee of Science and Technology (Grant No 10PJ1404400) and Nokia scholarship.

The modified context is then used in the filtering if it remains statistically significant in order to separate valid values from noisy ones. In this way, the context-based filtering is significantly improved.

2. PROPOSED METHOD

2.1 Context Tree Modeling

The classical context tree modeling technique has been widely used by data compression community with linear time complexity. A context 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 over each color for the current pixel relative to the node of context. Since not all possible contexts are present in the image, memory is only allocated for the actual number of pixel combinations. In our implementation, the spanning of tree is terminated once the frequency of the context on a given node is less than a predefined value.

2.2 Context Efficiency Validation

In image compression, all pixels must be encoded regardless of the reliability or probability of the context surrounding that pixel. One keeps track of how well it performs. In case of bad probability estimate, the coding of that pixel just takes more space. Optimal pruning of a context tree is always done on each of possible node in order to achieve a highest overall compression performance.

In image filtering, however, the main challenge is that the distribution of noise data is seldom known, and thus no evaluation can be done on how well filtering works. The critical issue is how to find some "meaningful" local contexts and filter only the pixels with lower conditional probability in local contexts. In DUDE [8], this decision rule is, in essence, a *MAP estimator*, which can be formulated as follows: for a local context **c**, the output of filtering x, equals to u_0, is the index value with highest conditional probability when equation (2) is met.

$$u_0 = \arg\max_{x \in A} P(x|\mathbf{c}) \tag{1}$$

$$\frac{(M-1)^2(1-\delta)}{\delta((1-\delta)M-1)} P(x = x_0|\mathbf{c}) - \frac{(M-1)}{((1-\delta)M-1)} P(x = u_0|\mathbf{c}) < 1 \tag{2}$$

Albeit DUDE has a so-called "*asymptotic optimality*" property, it demands an infinite sequence of data source for estimating all the conditional distributions of local contexts. This is of course not realistic in practice. In particular, when the context of a given pixel is contaminated by erroneous colors, the count corresponding to the symbol will be credited to the "wrong" context with rare appearance, which also causes inaccurate estimation of the context distribution.

This scenario motivates us to investigate a criterion for context classification. Those frequently contexts associated with a *dominant color* (conditional probability>90%), are termed as *good context*, on which filtering can be applied

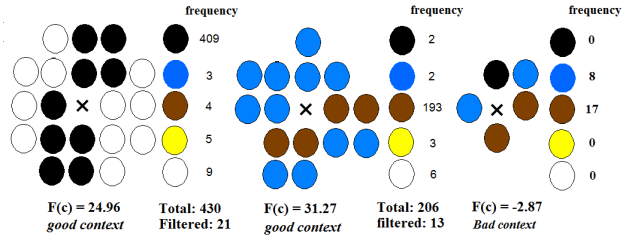

Fig. 1 *Examples of context classifications*

directly. Those rare appearance contexts, which include noise elements, we define as *bad* context because estimation of conditional probability under such contexts is inaccurate. In order to improve this estimation, a voting scheme is applied to find those noise pixels based in the *bad* contexts. An accurate conditional probability can be estimated on a new context excluding those noisy pixels. Here, all the contexts are categorized into three groups: *good*, *uncertain* or *bad* according to a so-called *context efficiency function*:

$$\mathrm{F}(\mathbf{c}) = \log_2\left(\frac{P(\mathbf{c})}{P_E(\mathbf{c})}\right) + k\sum_x P(x|\mathbf{c})\log\left(\frac{P(x|\mathbf{c})}{P(x)}\right) \tag{3}$$

where $P_E(\mathbf{c}) = \Pi_i P(y_i)$, $P(\mathbf{c})$ is the probability of a given context **c** and $P_E(\mathbf{c})$ is the estimated probability of **c**, y_i is the color of ith element in a given context **c**, and $P(y_i)$ the probability of y_i. $P_E(\mathbf{c})$ is computed by assuming all elements in the context are mutually independent.

In a sense of image compression, the first term in the right hand side of (3) can be interpreted as the difference of the context code length achieved according to the actual context probability and the expected context probability. Higher value for this term indicates that context **c** is a repetitive structure, and thus, it can be used as a direct filter when *dominant color* exists. The second term is the so-called *Kullback-Leiber distance* between conditional probability and color probability of the entire image. Larger distance implies that more bits can be saved when context **c** is used in coding.

To this end, all contexts are categorized into three groups accordingly: *good:* $F(\mathbf{c})>T_{max}$ with a *dominant color*, *bad*: $F(\mathbf{c}) < T_{min}$ and *uncertain*: otherwise. Once all contexts have been categorized into these three classes, the context tree is processed by identifying the *good* and *bad* nodes. The offspring nodes of any *good* or *bad* node will be removed in the tree pruning using top-to-bottom tracing. T_{max}, T_{min} and k are adopted as values of 3, -0.5 and 0.5 in this work. Three context examples are shown in Fig. 1. Two *good* contexts are with *dominant colors* of black (left) and brown (middle) both having the conditional probability over 90%, which can be used for filtering directly, while the right one is a *bad* context containing noise pixel in it. Fig. 3 shows an example of *good* and *bad* context distribution in a test image. The unmarked pixels belong to an *uncertain* context.

2.3 Voting Image

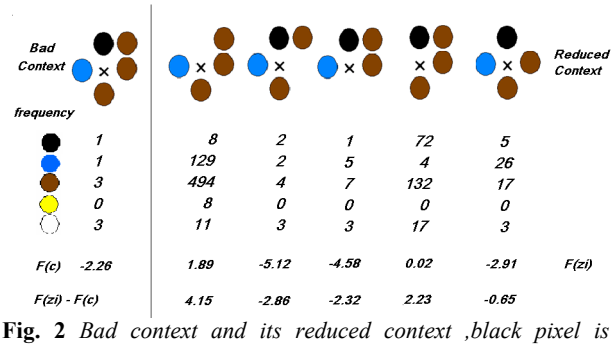

Fig. 2 *Bad context and its reduced context ,black pixel is possible a noise pixel with high $F(z_i)$ - $F(\mathbf{c})$ difference*

Since most of *bad* contexts contain noise pixels in themselves, they are seldom used to estimate a statistical model. However, they can be very useful in detection of noise pixels – except the case when a noisy pixel is isolated from most of inherited geometrical structures. Given an *bad* context **c**, a set of sub-contexts,

$$S(\mathbf{c}) = \{\mathbf{z}_i \mid \mathbf{z}_i = \mathbf{c}/\{x_i\}\}_{i=1}^{k} \tag{4}$$

are constructed by removing i^{th} pixel x_i from the context **c**, $i = 1, \ldots k$, where $\mathbf{z}_i$ is the reduced-size context after removal of the ith pixel. If the removed pixel x_i is a noise, it is expected that the reduced-size context will have a higher efficiency: $F(\mathbf{z}_i) > F(\mathbf{c})$, such that each pixel in **c** can be assigned with a meaningful value $F(\mathbf{z}_i) - F(\mathbf{c})$. The higher the difference, the more likely x_i is a noise pixel. Fig. 2 gives an example of *bad* context and its *reduced context*.

A *voting image R* is constructed according to the following rule: if the surrounding context **c** for a given pixel x is detected as *bad*, the accumulated voting score of each other pixel x_i in the same context **c** can be updated by:

$$R(x_i) = R(x_i) + F(\mathbf{z}_i) - F(\mathbf{c}) \tag{5}$$

Intuitively, we may conclude that most of contexts containing a noise pixel may be detected as *bad* contexts. If the noise pixel is removed from the *bad* context, the reduced-size context may have much better context efficiency. Namely, the accumulated voting score according to (5) is significantly higher than those of its neighborhood pixels. In this sense, the noise pixel can then be detected by finding the high peak points in the voting image. An example of a voting scheme is shown in Fig. 4.

2.4 Adaptive Context Selection

Once the voting image has been obtained, an adaptive context-based filter is applied in two manners. Firstly, if the surrounding context **c** in the context tree is *good*, there always exist one *dominant color* in the context, and eventually the value of the pixel x is replaced by the *dominant color* in that particular context if (2) meets. Secondly, if x is detected as a noise pixel in voting scheme and its surrounding context **c** is not labeled as *good*, the context is re-selected adaptively excluding those noise pixels using a 3×3 context template. Statistical distribution

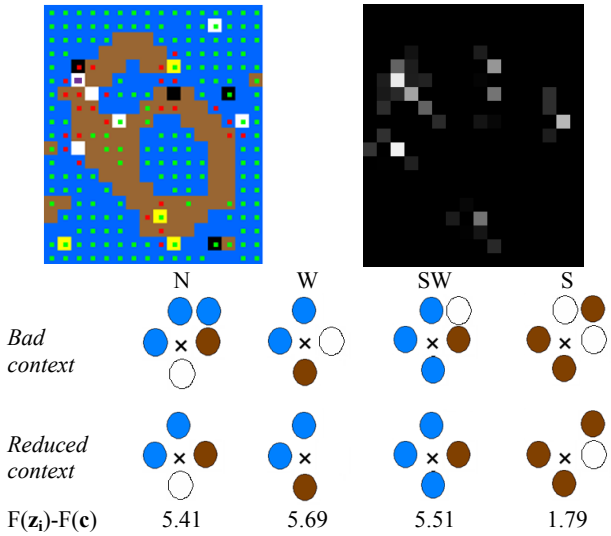

Fig. 3 *Sample image with good and bad context demonstrated in red and green colors (top left), its voting image (top right). Voting example for the white pixel labeled with purple with accumulated $F(z_i)$-$F(\mathbf{c})$ value 18.30 (bottom).*

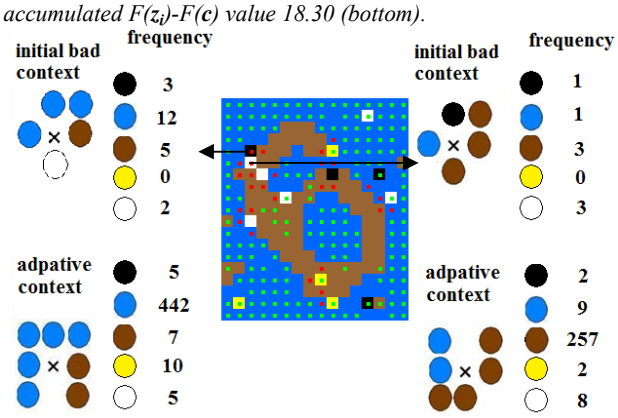

Fig. 4 *Example of adaptive context selection. For noise pixels (black and white, with high voting value), a new context in 3×3 region excluding surrounding noise pixels are selected, statistical information are collected for new contexts, black pixel is correctly changed to blue while white pixel changed to brown.*

of this adaptive context is collected and the DUDE framework is then applied. An example of adaptive context selection can be seen in Fig. 4. In order to improve the filtering robustness under different noise level, an estimation of δ is needed. Here, it is estimated by the minimum conditional probability occurred for contexts with "sufficient frequency", which is formulated as:

$$\delta = 1 - \max_{\forall x, p(\mathbf{c}) > 10^{-2}} P(x \mid \mathbf{c}) \tag{6}$$

The filtering can also be applied iteratively. Contrary to conventional filters, the images will not be degraded after applying several iterations.

2.5 Extension for Additive Gaussian Noise

During the map digitization, as limited color output is desired, a color quantization process can be applied for

reducing the number of colors. However, the noise will be incurred during the scanning process, which causes the possible overlapping in color space for some color components. Thus, we need to avoid misclassifying pixels caused by conventional color quantization. A novel iterative algorithm is proposed in [13] to optimize both the estimation of the indexed image and its color palette. For each pixel x, both the distance between RGB color vector to its corresponding component in the color palette, and its conditional probability of local context (estimated in Section 2.2-2.4) are taken into account as follows:

$$I(x) = \arg\min_{x=(1..M)}(-\log_2 f(\mathbf{y_x} \mid \mathbf{M}_x) - \log_2 P(x \mid \mathbf{c}))$$
$$f(\mathbf{y_x} \mid \mathbf{M_x}) = \exp(-\| \mathbf{y_x} - \mathbf{m_x} \|^2 / 2\sigma^2) \quad (7)$$

where $\mathbf{M_x}$ is 3-D *Gaussian distribution* with mean $\mathbf{m_x}$ and covariance matrix $\sigma^2\mathbf{I}$. This formula is similar to the energy function in *Markov random fields*, but the term *neighborhood homogeneity* is replaced by the conditional probability of the local context. The color components in color palette and the mean variance are updated following with the minimization step, see [13] for more details.

3. EXPERIMENTS

We evaluate the proposed *adaptive context-based filtering algorithm* (ACF) on a set of images from *National Land Survey of Finland*. For testing the performance of the filter, we artificially distort those images by adding impulsive or additive Gaussian noise. For comparison, four alternative filters [2, 3, 8, 10] are investigated for *impulsive noise* and four for *Gaussian noise* [4-7]. Performance comparisons are demonstrated in Fig. 5. It can be observes that the proposed filtering algorithm achieves significantly better numerical and visual quality.

4. CONCLUSION

We have proposed an efficient adaptive filtering using optimal context selection, which is designed via a novel voting-based noise estimation method. The proposed context-based filter can be viewed as a pilot study to conquer the raster map image distortion caused by uncertain noises. This algorithm can also be applied in other problem domains, such as image segmentation and color quantization.

5. REFERENCES

[1] M. Chen, M. Xu and P. Fränti, "Multi-layer filtering approach for map images", *IEEE Int. Conf. on Image Processing (ICIP'09)*, pp. 3953-3956, 2009.

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

[3] B. Smolka, A. Chydzinski, "Fast detection and impulsive noise removal in color images", *Real-Time Imaging*, 11(5-6), pp. 389-402, 2005.

[4] J. Portilla, V. Strela, M. Wainwright and E. P. Simoncelli, "Image denoising using a scale mixture of Gaussians in the wavelet domain," *IEEE Trans. on Image Proc,* 12(11), pp. 1338–1351, 2003.

[5] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising", *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 60-65,2005.

[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering", *IEEE Trans. on Image Proc*, 16(8), pp.2080–2095, 2007.

[7] A. Barbu. "Learning Real-Time MRF Inference for Image Denoising", *CVPR 2009.*

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

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

[10] E. Ordentlich, M. J. Weinberger, and T. Weissman, "Multi-directional context sets with applications to universal denoising and compression", *In Proc. of the 2005 IEEE Intl. Symp. on Inform. Theory, (ISIT'05)*, pp. 1270-1274, Sept. 2005.

[11] J. Yu and S. Verd´u, "Schemes for bidirectional modeling of discrete stationary sources", *IEEE Trans. Inform. Theory*, 52(11), pp.4789–4807,2006.

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

[13] M. Chen, M. Xu and P. Fränti, "Statistical filtering of raster map images", *IEEE Int. Conf. on Multimedia & Expo (ICME'10)*, pp. 394-399, 2010.

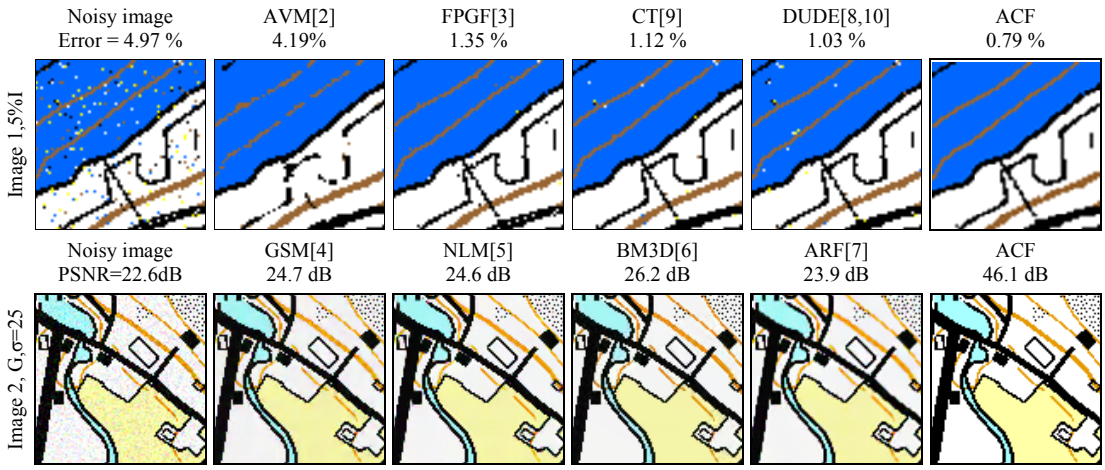

Fig. 5 *Performance comparison*