

TREE-SERIAL PARAMETRIC DYNAMIC PROGRAMMING WITH FLEXIBLE PRIOR MODEL FOR IMAGE DENOISING

Pham Cong Thang^{1,2}, Andrei V. Kopylov³

¹National Research University Higher School of Economics, 20 Myasnitskaya Street, Moscow, Russia

²The University of Da Nang – University of Science and Technology, 54 Nguyen Luong Bang Street, Da Nang, Viet Nam

³Tula State University, pr. Lenina 92, Tula, Russia

Abstract

We consider here image denoising procedures, based on computationally effective tree-serial parametric dynamic programming procedures, different representations of an image lattice by the set of acyclic graphs and non-convex regularization of a new type which allows to flexibly set a priori preferences. Experimental results in image denoising, as well as comparison with related methods, are provided. A new extended version of multi quadratic dynamic programming procedures for image denoising, proposed here, shows an improved accuracy for images of a different type.

Keywords: Image denoising, Dynamic programming, Bayesian optimization, Markov random fields (MRFs), Gauss-Seidel iteration method.

Citation: Thang PC, Kopylov AV. Tree-serial parametric dynamic programming with flexible prior model for image denoising. *Computer Optics* 2018; 42(5): 838-845. DOI: 10.18287/2412-6179-2018-42-5-838-845.

Acknowledgments: The results of the research project are published with the financial support of Tula State University within the framework of the scientific project № 2017-21-PUBL.

Introduction

A low-level processing is an important preliminary step in almost every image analysis system. Images may be corrupted by a disruptive noise during acquisition and transmission process. Thus, one of the main objectives of this low-level processing is to suppress noise, taking into account the essential requirement to preserve the local image structure for accurate and efficient subsequent analysis. The practical importance of noise suppression for image processing and computer vision keeps up a constant attention from the scientists. Many methods for edge-preserving image denoising [1 – 7] have been proposed in the literature, but none of them can simultaneously achieve both sufficient accuracy to provide a highly reliable data and computational speed to process super-resolution or dynamic images at a practice-relevant time. Therefore, denoising is still a challenging problem.

We apply here Bayesian approach as, perhaps, one of the most conceptually attractive and popular approaches to image processing. In the Bayesian framework, the problem of image denoising can be expressed as the problem of estimation of a hidden Markov component $\mathbf{X} = (x_t, \mathbf{t} \in T)$, $T = (\mathbf{t} = \{t_1, t_2\}, t_1 = 1 \dots N_1, t_2 = 1 \dots N_2)$ of a two-component (\mathbf{X}, \mathbf{Y}) Markov random field (MRF), where the analyzed noisy image $\mathbf{Y} = (y_t, \mathbf{t} \in T)$ plays the role of the observed component.

In the case of a singular loss function, the Bayesian estimation of the component \mathbf{X} can be found as a maximum a posteriori probability (MAP) estimation. It leads to the minimization problem of the objective function of a special kind related to the equivalent representation of Markov random fields in the form of Gibbs random fields in accordance with Hammersley–Clifford Theorem [8]. This function is often called the Gibbs energy function [9] and is formed as a sum of elementary objective functions of one or two variables associated, respectively, with nodes and edges of the pixel neighborhood graph.

The undirected graph that reflects occurring of pairs of variables in the given pairwise separable objective function is called the adjacency graph. Functions of one or two variables represent Gibbs potentials on cliques of adjacency graph and will be called node and edge functions correspondingly.

The edge functions are used to define the prior model of a hidden field \mathbf{X} and have a crucial influence on the edge-preserving properties. Different types of prior assumptions result in different types of edge functions. Various non-convex pairwise edge functions are considered in the literature [9-11] for preserving significant differences of values in the corresponding pairs of adjacent elements and smoothing the other differences. Convex edge-preserving pairwise potential functions were proposed to avoid the numerical involutions arising with non-convex regularization. However, non-convex regularization offers the best possible quality of image reconstruction with neat and exact edges [10]. One of the main problems in these approaches is the high computational complexity of corresponding minimization procedures which can hardly be applied to high-resolution images.

In this work, we develop parametric procedures for edge-preserving in image denoising based on dynamic programming (DP) principle, which consists in a recurrent decomposition of the initial problem of minimizing multivariate function into a succession of partial problems, each of which consists in minimizing a function of only one variable. These functions of one variable are called Bellman functions [12].

When the pairwise Gibbs potentials are selected as a minimum of a finite set of quadratic functions, and node functions are in quadratic form, the Bellman functions at each step of the dynamic programming will have a form of a minimum of a finite set of quadratic functions [11, 13, 14]. This lets us keep in memory and recalculate the parameters of these quadratic functions only, instead of

using discrete variables and calculation of the DP table, and therefore to reduce the overall computational time.

If the adjacency graph is a tree, then we have a version of a serial dynamic programming procedure [15] namely the tree-serial dynamic programming [16]. For images, the general form of the objective function will be the pairwise separable function with lattice-like adjacency of variables, so the tree-serial DP cannot be applied immediately to the image processing problems.

In our previous works, two different forms of tree-like representation of a rectangular lattice were utilized: tree-like approximation by the succession of simple trees [12], and the full set of possible acyclic adjacency graphs [17, 18]. The procedures can effectively remove an additive white Gaussian noise with high quality. Nevertheless, in both methods, the final decision for each variable is made based on a separate graph or the set of graphs. This can lead to inconsistency of the solution with the initial set of prior preferences. To overcome this obstacle, we propose an extended version of a multi quadratic dynamic programming procedure for image denoising based on the partitioning of the initial lattice on the non-overlapping set of chain-like graphs and iterative evaluation of separate groups of goal variables related to these graphs in accordance to the Gauss-Seidel method.

The experimental results show that the method usually converges within five-six iterations and exhibits the clear improvements in the denoising quality. Additionally, we have provided a comparison of the proposed approach with the other procedures for edge-preserving image denoising such as nonlinear total variation (TV) [4], modified Perona–Malik (MP-M) [7], Beltrami [19].

Related work

Within the Bayesian framework, the final decision about the hidden component $\mathbf{X}=(x_t, \mathbf{t} \in T)$ of a two-component (\mathbf{X}, \mathbf{Y}) MRF is based on posterior probability density distribution. The principle of average risk minimization in combination with singular loss function leads to the wide-spread decision rule, known as maximum a posteriori probability (MAP) estimation [9].

It is important that the pixel grid of an image is naturally supplied by the binary neighborhood relation, which turns it into a graph. The simplest case of such a graph is a rectangular lattice $G=(T \times T)$ (Fig. 1) which considers pairwise relations only and therefore has the clique number equal to two. In this case, the objective function for MAP estimation in the form of so-called Gibbs energy [9], will be the sum of functions of no more than two variables namely, the node and the edge ones [14]:

$$J(\mathbf{X} | \mathbf{Y}) = \sum_{\mathbf{t} \in T} \psi_{\mathbf{t}}(x_{\mathbf{t}} | \mathbf{Y}) + \sum_{(\mathbf{t}', \mathbf{t}'')} \gamma_{\mathbf{t}', \mathbf{t}''}(x_{\mathbf{t}'}, x_{\mathbf{t}''}). \quad (1)$$

In the papers [11, 14] we proposed to use a non-convex type of pairwise potential functions, that allow to flexibly set a priori preferences, using distinct penalty coefficients for various ranges of differences between the values of adjacent image elements:

$$\gamma_{\mathbf{t}'}(x_{\mathbf{t}'}, x_{\mathbf{t}''}) = u \min[\gamma_{\mathbf{t}'}^{(1)}(x_{\mathbf{t}'}, x_{\mathbf{t}''}), \dots, \gamma_{\mathbf{t}'}^{(L-1)}(x_{\mathbf{t}'}, x_{\mathbf{t}''}), \Delta^2] \quad (2)$$

where Δ, u – smoothing parameters, L – a number of quadratic functions,

$\gamma_{\mathbf{t}'}^{(i)}(x_{\mathbf{t}'}, x_{\mathbf{t}''}) = \lambda^{(i)}(x_{\mathbf{t}'} - x_{\mathbf{t}''})^2 + d^{(i)} -$ quadratic functions with parameters $\lambda^{(i)}$ и $d^{(i)}, i=1, \dots, L-1, (\mathbf{t}', \mathbf{t}'') \in G$.

These pairwise potential functions have the desired properties for the conservation of abrupt changes in the analyzed data.

Node functions are chosen in the quadratic form:

$$\psi_{\mathbf{t}}(x_{\mathbf{t}}) = (x_{\mathbf{t}} - y_{\mathbf{t}})^2. \quad (3)$$

When the pairwise Gibbs potentials are selected as a minimum of a finite set of quadratic functions (2), and node functions are in quadratic form (3), the Bellman functions at each step of the dynamic programming will also be a minimum of a finite set of quadratic functions. Therefore, each Bellman function belongs to the same parametric family, which leads to the parametric procedure called multi quadratic dynamic programming procedure (MQDP) [11, 13, 14]. The procedure searches for the minimum of the objective function (1) in two passes according to forward and backward recurrent relations [13, 14].

In the case of the graph G is a chain, the forward recurrent relation the Bellman function will be defined [12] in the following way:

$$\tilde{J}_t(x_t) = \psi_t(x_t) + \min_{x_{t-1}} [\gamma_t(x_{t-1}, x_t) + \tilde{J}_{t-1}(x_{t-1})], \quad (4)$$

$$t = 2, \dots, N, \quad \tilde{J}_1(x_1) = \psi_1(x_1).$$

$$\text{Let } \tilde{F}_t(x_t) = \min_{x_{t-1}} [\gamma_t(x_{t-1}, x_t) + \tilde{J}_{t-1}(x_{t-1})] \quad (5)$$

be a partial Bellman function. We have:

$$\tilde{J}_t(x_t) = \psi_t(x_t) + \tilde{F}_t(x_t).$$

The global minimum of the Bellman function (4) of the last variable $\min_{x_N} \tilde{J}_N(x_N)$ coincides with the global minimum of the total objective function for all variables:

$$\hat{x}_N = \arg \min_{x_N} \tilde{J}_N(x_N).$$

The other variables can be found by applying backward recurrent relation [12]:

$$\hat{x}_{t-1}(x_t) = \arg \min_{x_{t-1}} [\gamma_t(x_{t-1}, x_t) + \tilde{J}_{t-1}(x_{t-1})].$$

The amount of quadratic functions grows at each step of the forward move but not all of them take part in forming the minimum. These functions can be dropped using simple enough procedure that considers the position of the minimum point and the points of intersection of quadratic functions with each other (Algorithm 1).

It was proven, that in the case of signal processing the number of quadratic functions \tilde{L}_t , that are required for representation of a Bellman function $\tilde{J}_t(x)$, generally, is not increased by more than one at each step.

For images, the general form of the objective function will be the pairwise separable function with lattice-like adjacency of variables (Figure 1).

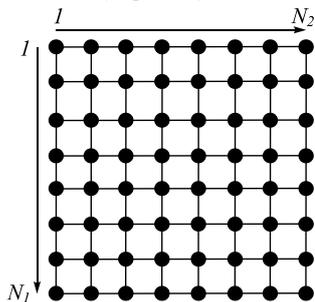


Fig. 1. Neighborhood graph on the set of data array elements: a rectangular lattice

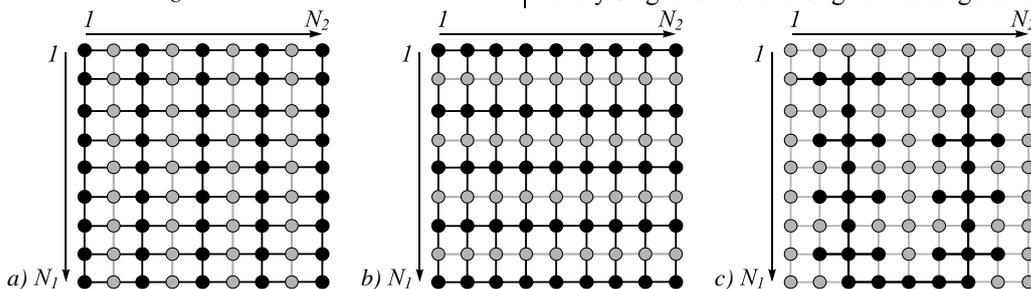


Fig. 2. Examples of the decomposition of a lattice-like adjacency graph into the set of subtrees: (a) odd and even columns, (b) odd and even rows, (c) more complex decomposition into two trees

For this combination of partial pixel neighborhood trees, the algorithm of finding the optimal values of the stem-node variables boils down to a combination of two usual DP procedures, dealing with single, respectively, horizontal and vertical rows of the image.

Hereinafter we will call this procedure the tree approximation algorithm.

We use a separate pixel neighborhood tree, which is defined, nevertheless, on the whole pixel grid and has the same horizontal branches as the others. The resulting image processing procedure is aimed at finding optimal values only for the hidden variables at the stem nodes in each tree. For this combination of partial pixel neighborhood trees, the algorithm of finding the optimal values of the stem-node variables boils down to a combination of two usual dynamic programming procedures, each deal with single, respectively, horizontal and vertical image rows considered as signals on the one-dimensional argument axis [20, 21]. First, such a one-dimensional procedure is applied to the horizontal rows independently for each column to find so-called marginal functions $\tilde{J}_t(x_t)$ which represent the minimum of partial criterion, associated with the particular row with respect to all variables besides x_t [22]:

$$\tilde{J}_t(x_t) = \tilde{F}_t^-(x_t) + \tilde{F}_t^+(x_t) + \psi_t(x_t), \tag{6}$$

where

$$\tilde{F}_t^-(x_t) = \min_{x_{t-1}} [\gamma_t(x_{t-1}, x_t) + \tilde{F}_{t-1}^-(x_{t-1}) + \psi_{t-1}(x_{t-1})],$$

and

$$\tilde{F}_t^+(x_t) = \min_{x_{t+1}} [\gamma_t(x_t, x_{t+1}) + \tilde{F}_{t+1}^+(x_{t+1}) + \psi_{t+1}(x_{t+1})].$$

A lattice is not a chain, and Bellman’s DP principle cannot be immediately applied to solving such optimization problems. There are, at least, four ways [16, 17] of using computational advantages of the tree-serial DP for optimization of an objective function with a lattice-like graph of variable adjacency (Figure 2).

The first, and the most often applied to various image processing problems technique is based on the following heuristic idea [12]. We can decompose the original lattice-like adjacency graph into several tree-like ones each of which covers, nevertheless, all the elements of the pixel grid. Respectively, instead of the overall objective function, we use several partial functions with tree-like adjacency of variables to evaluate the goal variables at every single row of the original rectangular lattice.

Then, the procedure is applied to the vertical rows independently for each row with the only alteration: the respective marginal node functions $\tilde{J}_t(x_t)$, obtained at the first step, are taken instead of the image-dependent node functions $\psi_t(x_t)$. In the case of real-valued variables x_t and quadratic pairwise separable objective function, these elementary procedures applied to single horizontal and vertical rows are nothing else than Kalman filters-interpolators [23] of a special kind. Nevertheless, using MQDP for processing two-dimensional data based on the tree approximation of lattice-like neighborhood graph, the number of quadratic functions in the Bellman functions may be too large and leads to a lack of effective implementation of the procedure. In paper [11, 14] the following heuristic technique was proposed to preserve the general form of a Bellman function with several minimums. The idea was to divide the quadratic functions into groups. The number of groups is determined by the number of stored minimum values. For the separation of functions into the group, we used the featureless modification of the most popular clustering algorithms – k-means, which possesses the benefits of speed and ease of implementation [14].

In the paper [17] another way for the tree-like approximation of a lattice based on the full set of acyclic adjacency graphs was proposed. Let a hypothetical covering set of all spanning acyclic graphs (the full set) be given. For the finite set of image elements, the number of such graphs is also finite. Let us assume that all elements of the data array be roots for several unknown for us acyclic adjacency graphs from the full set. Expanding step-by-step vicinities of descendants for each element

simultaneously, we can obtain its maximal vicinity including the element itself, and thus obtain the final decision for that element as a combination of decisions based on acyclic adjacency graphs from the full set. The paper describes the general probabilistic framework for dependent objects recognition.

The third way [21] consists in finding such a cut of the graph G that it gets resolved into trees G^1, G^2, \dots, G^K ; $G = \bigcup_{j=1}^K G^j$, $G_i \cap G_j = \emptyset$, $i \neq j$.

Then, it will be possible to apply the Gauss-Seidel iteration method to optimize the entire objective function by finding, at each step, the global optimum of each of the partial objective functions on trees G^i , $i = 1, \dots, K$ using tree-serial DP.

Algorithms based on the iterative reevaluation of groups of variables are significantly more robust to local extrema.

The fourth technique considers the lattice as a chain of rows, so, we can consider the variables in each row as an aggregated variable and apply the serial dynamic programming technique which is a particular case of tree-serial dynamic programming [24]. However, the Bellman functions will be functions of as many variables as long are the rows. Nevertheless, the computational complexity of such a procedure will be linear with respect to the number of rows, remaining, though, very high relative to their length. To further reduce the computational complexity of optimization, we heuristically approximate the genuine Bellman functions by appropriate pairwise separable substitutes. However, it is hard to find an appropriate approximation in the case of an edge-preserving Gibbs potentials and we cannot apply this way of approximation in our case.

Approximation algorithms allow constructing the multi quadratic dynamic programming procedures for image processing on the basis of tree-approximation and the full set of adjacency graphs [17, 25]. The second procedure utilizes an increased accuracy of the tree-like representation of a lattice based on the full set of adjacency graphs, described in [18], and let us to sufficiently simplify the recalculation of the intermediate Bellman functions.

Nevertheless, in both methods, the final decision is made based on a separate graph or the set of graphs. This can lead to inconsistency of the solution with the initial set of prior preferences. A construction of an enhanced version of MQDP based on the partitioning of the initial lattice on the set of non-overlapping set of chain-like graphs and iterative evaluation of separate groups of goal variables related to these graphs in accordance to the Gauss-Seidel method can improve the overall accuracy of image denoising.

Iterative multi-quadratic procedure

As described in the previous section, we can find such a cut of the graph G that it gets resolved into two trees. Then, it will be possible to apply the Gauss-Seidel iteration method to optimize the entire objective function by finding, at each step, the global optimum of each of the two-partial objective function using tree-serial dynamic programming. One of the simplest ways to divide a lattice graph into trees is to combine all the variables into two groups, namely the variables corresponding to the odd

and even rows of the original lattice. Moreover, a chain-like decomposition let us skip the k-means clustering step of reducing the number of quadratic functions in the representations of Bellman functions [11, 14], as the node functions remain in quadratic form at each iteration step. This can greatly reduce the computation complexity of the procedure.

Further, in accordance with the Gauss-Seidel principle, starting from some initial approximation, we carry out optimization with respect to the first group of variables, for fixed values of the variables of the second group. Then we carry out optimization with respect to the variables of the second group, with the fixed values of the first group found in the previous step, etc. until the procedure converges, and the local minimum of the criterion is reached.

As before [11, 14, 18] we will use Algorithm 1 to select quadratic functions in the representation of the Bellman functions at each step of dynamic programming.

Algorithm 1:

Reduction of a number of quadratic functions

Input: $\tilde{F}_t^{(i)}(x_t), i = 1, \dots, \tilde{K}_t$.

Output: $\tilde{F}_t^{(i)}(x_t), i = 1, \dots, \tilde{K}'_t, \tilde{K}'_t < \tilde{K}_t$

1. In the beginning, sort by ascending values $\tilde{d}_t^{(i)}$ of the array $\tilde{F}_t^{(i)}(x_t)$.
 2. In case of the presence of a parameter Δ at each step, we look for the minimum constant $D_t = \tilde{d}_t^{(1)} + u \cdot \Delta^2$ and reject all other constants.
 3. Discard all functions that have minimum greater than or equal to this constant $\tilde{d}_t^{(i)} \geq D_t, i = 2 \dots \tilde{K}_t$.
 4. Among the remaining functions, select the function with the smallest minimum and keep it.
 5. Find and check the necessary and sufficient condition of the intersection of quadratic functions by equations. Discard all the functions for which there is no intersection.
 6. Among the remaining functions, select the function with the smallest minimum and keep it.
 7. Repeat until there exist functions for which no decision on acceptance or discarding is done.
-

The proposed iterative multi-quadratic procedure for image denoising is described in Algorithm 2.

However, the Gauss-Seidel method guarantees to find the global optimum only if the objective function is convex or concave. But the described above procedure can improve the quality of initial approximation obtained by means of noniterative multi quadratic dynamic programming procedure. Thus, the general algorithmic scheme of solving pairwise separable optimization problem consists in two steps. At first, the initial approximation must be found by means of the fast tree approximation algorithm, and then the iterative tree-decomposition procedure starts from this approximate solution.

To overcome the locality effect of the Gauss-Seidel search, the can change the way of decomposition as soon as the next local minimum is achieved. The computational complexity of each iteration is linear relative to the number of pixels.

The initial approximation $\mathbf{X}^* = (x_t^*, \mathbf{t} \in T)$ could be chosen in different ways. For example, we can start from the observable image itself, or use the results of tree-decomposition based procedure [11] or procedure on the full set of adjacency graphs [18] as the initial approximation for faster convergence.

Algorithm 2: Iterative multi-quadratic procedure

Input: Observed image $\mathbf{Y} = (y_t, \mathbf{t} \in T)$,
 Initial approximation: $\mathbf{X}^* = (x_t^*, \mathbf{t} \in T)$.
MaxIter, *oddness* = 0.
Output: Resulting image $\mathbf{X} = (x_t, \mathbf{t} \in T)$.

- o while k=1 to **MaxIter** do
 - o $\mathbf{t} = \mathbf{t} + \text{oddness}$
 - o $\phi_t(x_t) = (x_t - y_t)^2 + \gamma(x_t, x_{t+1}^*) + \gamma(x_t, x_{t-1}^*)$
 - o Calculate functions $\tilde{F}_t(x_t)$ by form (5) with node function $\phi_t(x_t)$ and applying Algorithm 1.
 - o Calculate marginal node functions $\tilde{J}_t(x_t)$ with node functions (3) by form (6) and applying Algorithm 1 without his steps 2-3.
 - o $\hat{x}_t(x_t) = \arg \min \tilde{J}_t(x_t)$
 - o $\mathbf{t} = \mathbf{t} + 2$
 - o $(\mathbf{X}^*(x_t^*) = \hat{\mathbf{X}}(\hat{x}_t))$
 - o *oddness* = 1 - *oddness*
- o Return $\hat{\mathbf{X}} = (\hat{x}_t)$

If we apply Algorithm 2 to the rows of the image, then, the $\mathbf{t} + \text{oddness} = \{t_1 + \text{oddness}, t_2\}$, x_{t-1} means x_{t_1-1, t_2} , and x_{t+1} corresponds to x_{t_1+1, t_2} . In the same time, we can apply Algorithm 2 to the columns of the image, when e.g. the next local minimum is reached. In this case $\mathbf{t} + \text{oddness} = \{t_1, t_2 + \text{oddness}\}$, x_{t-1} means x_{t_1, t_2-1} , etc.

The previously proposed procedures [11, 14, 18] on the basis of MQDP use different tree-like approximations of the lattice-like pixel grid of an image. Thus, such procedures do not search the minimum of the initial criterion (1). Instead, they form a set of partial criteria that is close in some sense to the initial and find the exact global minimum of this criteria.

In contrast, the procedure described here is intended to solve the initial optimization task but it can provide a local minimum only. Therefore it has different behavior in comparison with the methods [11, 18] and can achieve better results, as evidenced by experiments.

Experimental results

All the experiments were run on a machine with Core i7 – CPU 2GHz, SDRAM 4 GB, Windows 10 (64 bit), and implemented in Matlab.

The test images are 8-bit grayscale standard images with various levels of additive white Gaussian noise. Standard deviations are chosen to be $\sigma = 10, 20, 30$. Denoising results are quantified by the Mean Structure Similarity Index (SSIM) and Peak to Signal Noise Ratio (PSNR) [1].

Details of quantified measurements are shown in Table 1 with different Gaussian noise levels. Results of measurements in Table 1 are averaged over multiple tests.

Table 1. Performance of various methods as measured by PSNR and SSIM corresponding to varying noise levels

Image	House 128x128					
	10		20		30	
TV	28.74	0.8279	26.14	0.7440	24.06	0.6538
M-PM	30.37	0.8559	26.58	0.7337	24.40	0.6499
Beltrami	30.46	0.8695	26.63	0.7544	24.38	0.6679
MQDP	30.48	0.8708	26.60	0.7841	24.30	0.6886
Proposed Horizontal	30.58	0.8801	26.68	0.7949	24.41	0.7418
Proposed Vertical	30.54	0.8798	26.62	0.7930	24.33	0.7397
Proposed Combination	30.68	0.8809	26.72	0.7966	24.43	0.7429
Image	Lake 128x128					
TV	25.67	0.8947	23.14	0.8118	20.97	0.7173
M-PM	27.52	0.9183	23.35	0.8192	21.19	0.7359
Beltrami	26.32	0.8997	23.97	0.8438	21.61	0.7660
MQDP	29.54	0.9220	24.09	0.8450	21.27	0.7759
Proposed Horizontal	29.74	0.9268	24.37	0.8591	21.37	0.7923
Proposed Vertical	29.54	0.9250	24.17	0.8564	21.37	0.8003
Proposed Combination	29.84	0.9293	24.52	0.8721	21.79	0.8066
Image	Parrot 128x128					
TV	26.06	0.8458	24.30	0.7984	22.76	0.7298
M-PM	29.35	0.9041	25.90	0.7973	23.56	0.7072
Beltrami	28.68	0.8919	25.98	0.8146	23.49	0.7283
MQDP	29.64	0.8955	26.01	0.8380	23.37	0.7376
Proposed Horizontal	29.83	0.9087	26.15	0.8395	23.51	0.7864
Proposed Vertical	29.71	0.8977	26.09	0.8388	23.42	0.7834
Proposed Combination	29.84	0.9177	26.22	0.8411	23.70	0.7886
Image	Hill 256x256					
TV	30.30	0.8292	27.78	0.7409	25.97	0.6587
M-PM	30.55	0.8314	27.88	0.7470	25.69	0.6464
Beltrami	30.72	0.8402	28.00	0.7534	25.82	0.6544
MQDP	30.69	0.8579	27.98	0.7610	26.17	0.6842
Proposed Horizontal	30.74	0.8854	28.10	0.7985	26.32	0.7372
Proposed Vertical	30.86	0.8804	28.08	0.7916	26.28	0.7362
Proposed Combination	30.89	0.8865	28.16	0.8055	26.41	0.7381
Image	F16 256x256					
TV	30.22	0.8775	26.55	0.7434	24.12	0.6187
M-PM	30.29	0.8845	26.79	0.8005	24.64	0.6425
Beltrami	30.36	0.8896	26.74	0.8244	24.87	0.6815
MQDP	30.50	0.8913	26.88	0.8587	24.75	0.7204
Proposed Horizontal	30.90	0.8927	27.13	0.8688	24.88	0.7334
Proposed Vertical	30.82	0.8921	27.03	0.8668	24.86	0.7387
Proposed Combination	31.04	0.8944	27.18	0.8709	25.03	0.7420
Image	Bridge 256x256					
TV	27.40	0.7876	25.63	0.7183	24.16	0.6480
M-PM	29.74	0.8878	26.12	0.7753	24.27	0.6845
Beltrami	30.02	0.8908	26.23	0.7781	24.23	0.6860
MQDP	30.08	0.8982	26.15	0.7898	24.31	0.7079
Proposed Horizontal	30.16	0.9141	26.31	0.8005	24.38	0.7126
Proposed Vertical	30.11	0.9087	26.26	0.7993	24.42	0.7116
Proposed Combination	30.22	0.9159	26.32	0.8028	24.48	0.7140

In Fig. 3, we use probe ‘synthetic’ image with the size of 20x20 pixels to compare results of the multi-quadratic procedure (MQDP) [11] and proposed iterative multi-quadratic procedure method with iterative horizontal processing, iterative vertical processing, and its combination. A number of iteration steps is equal to 6.

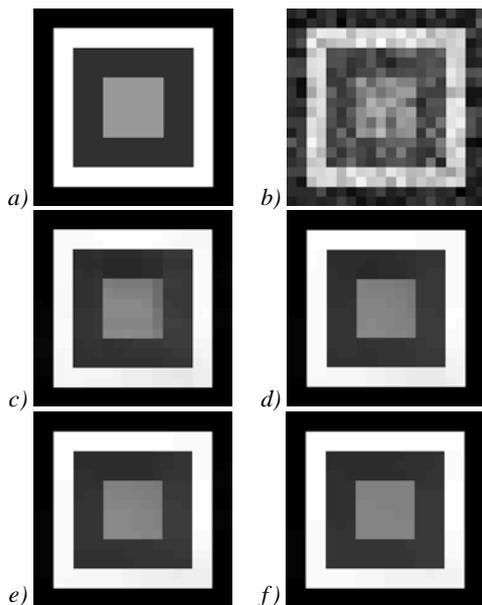


Fig. 3. a) Image “synthetic” with the size of 20×20 pixels;
 b) Noisy PSNR = 18.5907;
 c) MQDP: PSNR = 29.70, SSIM = 0.9784;
 d) Iterative horizontal: PSNR = 30.93, SSIM = 0.9862;
 e) Iterative vertical: PSNR = 31.51, SSIM = 0.9873;
 f) Combination: PSNR = 32.94, SSIM = 0.9906

For illustrations, we use the gray level images House, Lake, Parrot, Hill, F16, Bridge. The original images are presented in Figure 4.

Figure 5 and 6 show comparative result images of proposed and other methods for edge-preserving image denoising: Nonlinear Total Variation (TV) [4], Beltrami [19], Modified Perona–Malik (MP-M) [7], Multi quadratic dynamic programming (MQDP) [11].

For Beltrami method, the aspect ratio is set to 1, balancing parameter between fidelity and regularity is the inverse proportionality of noisy level, gradient descent parameter is set to 0.2, tolerance for convergence crite-

riion is equal to 2×10^{-5} and the maximum number of iterations is set to 2000.

For the M-PM method, the diffusivity parameter is 3, integration constant is set to 0.1 and the number of iterations is the direct proportionality of noise level.

For the TV method, the regularization parameter is set to 0.075, tolerance for convergence criterion is equal to 5×10^{-4} and the maximum number of iterations is set to 300.

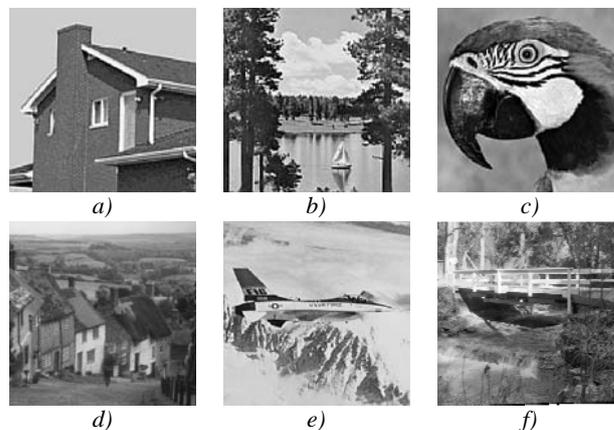


Fig. 4. Original images: a) House, b) Lake, c) Parrot, d) Hill, e) F16, f) Bridge

For MQDP method, we use edge functions (2) with fixed smoothing parameters values $L=3, \lambda=0.2, d=0.5\Delta^2$.

The PSNR and SSIM results for various methods are shown to demonstrate the comparison of their performances.

Conclusion

In this paper, we have tried to discuss promising edge-preserving image denoising methods on the basis of different principles of tree-like representation of an image lattice, and MQDP. A new MQDP version based on the tree-decomposition of the image pixel grid searches for the local minimum for the initial criterion (1), instead of the global minimum of its approximations.

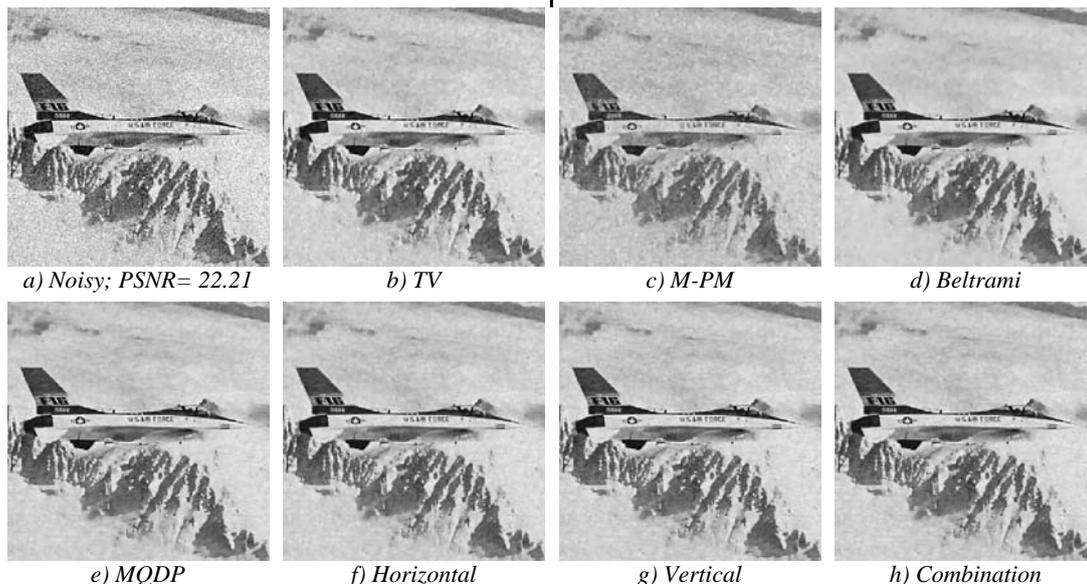


Fig. 5. Results for image “F16” (256×256): Image denoising results of the compared methods

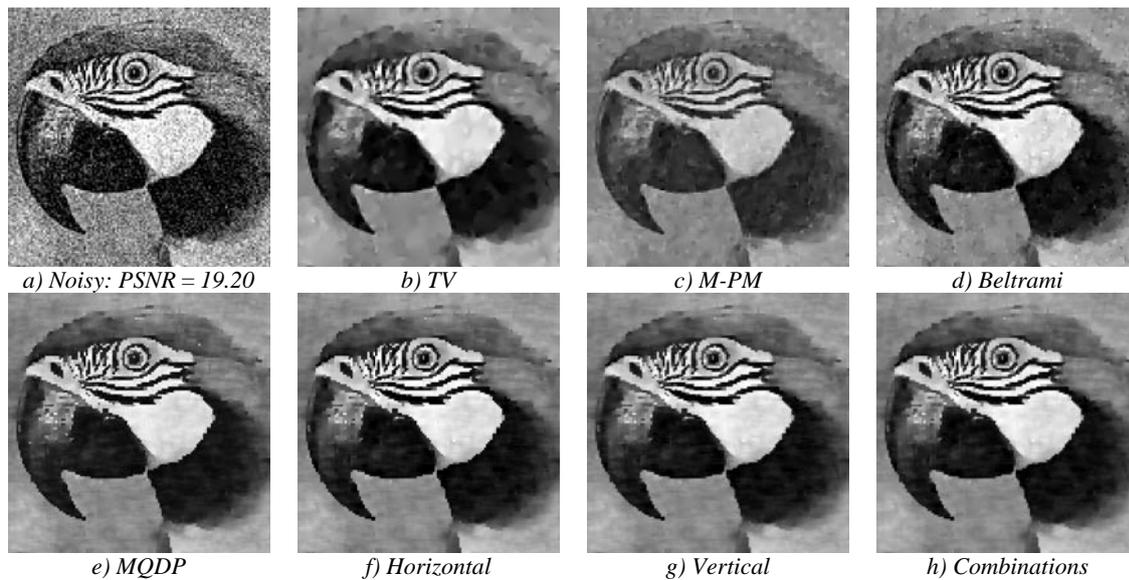


Fig. 6. Results for image "Parrot" (128×128): Image denoising results of the compared methods

The experimental results show that the proposed dynamic programming procedure allows improving the quality of results of multi quadratic dynamic programming. Numerical results show that our proposed algorithms are efficient and allow to obtain result compared or even superior the existed methods.

References

- [1] Wang Z, Bovik AC. Modern image quality assessment. Synthesis Lectures on Image, Video, and Multimedia Processing. Morgan & Claypool Publishers; 2006. ISBN: 978-1-59829-022-6.
- [2] Chambolle A, Caselles V, Novaga M. Total variation in imaging. In Book: Scherzer O, ed. Handbook of mathematical methods in imaging. New York: Springer Science+Business Media LLC; 2011: 1016-1057. DOI: 10.1007/978-0-387-92920-0_23.
- [3] Chaudhury K, Sage D, Unser M. Fast $O(1)$ bilateral filtering using trigonometric range kernels. IEEE Transactions on Image Processing 2011; 20(12): 3376-3382. DOI: 10.1109/TIP.2011.2159234.
- [4] Rudin LI, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena 1992; 60(1-4): 259-268. DOI: 10.1016/0167-2789(92)90242-F.
- [5] Wang Y, Chen W, Zhou S, Yu T, Zhang Y. MTV: modified total variation model for image noise removal. Electronics Letters 2011; 47(10): 592-594. DOI: 10.1049/el.2010.3505.
- [6] You Y-L, Kaveh M. Fourth order partial differential equations for noise removal. IEEE Transactions on Image Processing 2000; 9(10): 1723-1730. DOI: 10.1109/83.869184.
- [7] Wang YQ, Guo JC, Chen WF, Zhang W. Image denoising using modified Perona-Malik model based on directional Laplacian. Signal Processing 2013; 93(9): 2548-2558. DOI: 10.1016/j.sigpro.2013.02.020.
- [8] Hammersley JM, Clifford PE. Markov random fields on finite graphs and lattices. Berkeley preprint; 1971.
- [9] Besag JE. Spatial interaction and the statistical analysis of lattice systems. Journal of the Royal Statistical Society, Series B 1974; 36(2): 192-236.
- [10] Nikolova M, Michael K, Tam C-P. Fast nonconvex non-smooth minimization methods for image restoration and reconstruction. IEEE Transactions on Image Processing 2010; 19(12): 3073-3088. DOI: 10.1109/TIP.2010.2052275.
- [11] Pham CT, Kopylov AV. Parametric procedures for image denoising with flexible prior model. Proceedings of the Seventh Symposium on Information and Communication Technology (SoICT '16) 2016: 294-301. DOI: 10.1145/3011077.3011099.
- [12] Mottl VV, Blinov AB, Kopylov AV, Kostin AA. Optimization techniques on pixel neighborhood graphs for image processing. In Book: Jolion J-M, Kropatsch WG, eds. Graph-based representations in pattern recognition. Vienna: Springer-Verlag; 1998: 135-145. DOI: 10.1007/978-3-7091-6487-7_14.
- [13] Pham CT. Image processing procedures based on multi-quadratic dynamic programming. Informatica 2017; 41(2): 255-256.
- [14] Pham CT, Kopylov AV. Multi-quadratic dynamic programming procedure of edge-preserving denoising for medical images. ISPRS – International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 2015; XL-5/W6: 101-106. DOI: 10.5194/isprsarchives-XL-5-W6-101-2015.
- [15] Bertelè U, Brioschi F. On non-serial dynamic programming. Journal of Combinatorial Theory, Series A 1973; 14(2): 137-148. DOI: 10.1016/0097-3165(73)90016-2.
- [16] Mottl V, Kopylov AV, Kostin A, Yermakov A, Kittler J. Elastic transformation of the image pixel grid for similarity based face identification. Proceedings of 16th International Conference on Pattern Recognition 2002; 3: 549-552. DOI: 10.1109/ICPR.2002.1047998.
- [17] Dvoenko SD. Clustering sets based on distances and proximities between its elements [In Russian]. Sibirskii Zhurnal Industrial'noi Matematiki, 2009; 12(1): 61-73.
- [18] Pham CT, Kopylov AV, Dvoenko SD. Edge-preserving denoising based on dynamic programming on the full set of adjacency graphs. ISPRS – International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 2017; XLII-2/W4: 55-60. DOI: 10.5194/isprs-archives-XLII-2-W4-55-2017.
- [19] Zosso D, Bustin A. A primal-dual projected gradient algorithm for efficient Beltrami regularization. Tech Rep UCLA CAM Report 14-52 2014.

- [20] Kopylov AV. Parametric dynamic programming procedures for edge preserving in signal and image smoothing. *Pattern Recognition and Image Analysis* 2005; 15(1): 227-230.
- [21] Kopylov A. Tree-serial dynamic programming for image processing. *Proceedings of 19th International Conference on Pattern Recognition* 2008: 1-4. DOI: 10.1109/ICPR.2008.4761407.
- [22] Kopylov A, Krasotkina O, Pryimak A, Mottl V. A signal processing algorithm based on parametric dynamic programming. In Book: Elmoataz A, Lezoray O, Nouboud F, Mammass D, Meunier J, eds. *Image and Signal Processing*. Berlin, Heidelberg: Springer; 2010: 280-286. DOI: 10.1007/978-3-642-13681-8_33.
- [23] Kalman RE, Bucy RS. New results in linear filtering and prediction theory. *Journal of Basic Engineering* 1961; 83(1): 95-108. DOI: 10.1115/1.3658902.
- [24] Kopylov AV. Rowwise aggregation of variables in the dynamic programming algorithm for image processing. *Pattern Recognition and Image Analysis* 2008; 18(2): 309-313. DOI: 10.1134/S105466180802017X.
- [25] Dvoenko SD. Recognition of dependent objects based on acyclic Markov models. *Pattern Recognition and Image Analysis* 2012; 22(1): 28-38. DOI: 10.1134/S1054661812010130.

Author's information

Pham Cong Thang (b. 1988) graduated from Tula State University in 2013, received his Ph.D. in 2016. He is a lecturer at The University of Da Nang-University of Science and Technology, Da Nang, Viet Nam. Since 09.2017, he is a research fellow at National Research University Higher School of Economics with the support of The Russian "5-100" Academic Excellence Project. His research interests are image processing, machine learning. E-mail: pacotha@gmail.com, pcthang@dut.udn.vn.

Kopylov Andrei Valerievich (b. 1970) graduated from Tula State University in 1992, received his Ph.D. in 1997 and currently works at the Institute of Applied Mathematics and Computer Science of Tula State University as an associate professor. His scientific interests are data mining, optimization approach to signal and image analysis, statistical methods in image analysis, edge-preserving filtering of random fields etc. E-mail: And.Kopylov@gmail.com.

*Code of State Categories Scientific and Technical Information (in Russian – GRNTI): 28.23.15.
Received November 27, 2017. The final version – July 27, 2018.*
