Improved Recursive Geodesic Distance Computation for Edge Preserving Filter

Mikhail G. Mozerov, *Member, IEEE* and Joost van de Weijer, *Member, IEEE*

***Abstract*—All known recursive filters based on the geodesic distance affinity are realized by two 1D recursions applied in two orthogonal directions of the image plane. The 2D extension of the filter is not valid and has theoretically drawbacks which lead to known artifacts. In this paper a maximum influence propagation method is proposed to approximate the 2D extension for the geodesic distance based recursive filter. The method allows to partially overcome the drawbacks of the 1D recursion approach. We show that our improved recursion better approximates the true geodesic distance filter, and the application of this improved filter for image denoising outperforms the existing recursive implementation of the geodesic distance. As an application we consider a geodesic distance based filter for image denoising. Experimental evaluation of our denoising method demonstrate comparable and for several test images better results, than state-of-the-art approaches, while our algorithm is considerably faster with computational complexity O(8P).**

***Index Terms*—Geodesic distance filter, color image filtering, image enhancement.**

I. Introduction

The geodesic distance metric is popular in image processing and computer vision applications including segmentation [7], [26], object proposal computation [15], stereo estimation [17] and image filtering [12]. Originally, a geodesic is the shortest route between two points on the Earth surface. The geodesic distance is a generalization of the straight line distance in the Euclidean space to the distance measure in a curved space. In the case of images, the geodesic distance is defined as the shortest path on the surface between two points. Here the surface is formed by the image value function defined on the 2D spatial domain. In several applications the geodesic distance needs to be computed between a limited set of seed points for which very fast algorithms exist [7]. However, for other applications such as image filtering and energy minimization the geodesic distance needs to be computed between all possible pixel pairs in the image and the computational complexity of this kind of the filter is demanding. In this article we consider applications which require geodesic distance computation between all pairs of points in the image.

Known methods which address this problem approximate the geodesic distance recursively by applying the 1D recursion twice, firstly along the vertical coordinate and then along horizontal coordinate (or vice versa) of the image plane (abbreviated with 1D recursion in the remainder of the paper or with XY and YX if the order of the consecutive recursions is known). One of the first attempts to apply the filter with the geodesic distance based kernel was proposed in [31], and later in [8]. A theoretical basis for edge-preserving filtering with geodesic distance has been proposed in [12] and further extended in [29]. Note that the geodesic distance based filters belong to the more wide class of the edge-preserving smoothing filters [4], [24], [25] and have received considerable attention in image processing, computer graphics, and computer vision. The filter has been applied to a wide variety of applications such as edit propagation [2], denoising [5], stereo matching [17], [19], and optical flow [28], [18], video abstraction and demosaicing [23], [27].

However the 1D recursion has several drawbacks. The consecutive application of the filter along X and Y produces stripe-like artifacts [12], which can be suppressed only by iterative application with an unpredictable number of iterations. The second drawback of the 1D recursion is that filter is not rotation invariant. In Section 2 we explain these artifacts in more details.

The proposed maximum influence propagation algorithm approximates the true geodesic distance (our approach is abbreviated with 2D). The idea is to choose a locally optimal path between two neighbor directions by maximizing the integral influence. Note that for the filters that are based on geodesic distance the approximation of the weights (which values are exponentially inverse to the corresponding geodesic distances) of the local neighborhood pixels is of more importance than the actual distances itself. And these influence weights form support regions around filtered pixels. This is why we approximated the pixel-wise integral influence instead of the true geodesic distances. We will evaluate the geodesic distance approximation for the task of image denoising. We show that results of this filter can be greatly improved by pre-filtering the image from which the affinity space function is computed. Our denoising algorithm considerably outperforms state-of-the-art method for several piecewise constant like images (House and Cameraman) and shows comparable results for other test images, while being considerably faster with computational complexity O(8P).

This paper is organized as follows in Section II we propose our new approach to geodesic distance computation for the application of image filtering. In Section III we describe our denoising algorithm. In Section IV we present the experimental results, and we conclude in Section V.

M. Mozerov and J.van de Weijer are with the Computer Vision Center of Department Informatics, Universitat Autnoma de Barcelona, Barcelona, Spain, 08193.
E-mail: mozerov@cvc.uab.es

II. Fast Geodesic distance computation for image filtering

We first address geodesic distance computation within the context of image filtering, where it has been principally used for fast bilateral filter computation. The classic bilateral kernel is a function of the Euclidean distance in the joint color and spatial coordinates multidimensional space. The computational complexity of the brute-force implementation for this kind of kernels is highly demanding. Several fast algorithms were proposed in recent years [1], [13], [20], [21], [22], where the approximation achieves high quality. The computational complexity for the fast realization usually depends on filter parameters. Thus, the geodesic distance based recursive filter is a real alternative to the bilateral filter due to its natural ability to be calculated recursively and the constant computational complexity, which does not depend on any intrinsic parameters. Actually, the work of Yang [29] on recursive bilateral filtering proposed to use the geodesic distance for its computation.

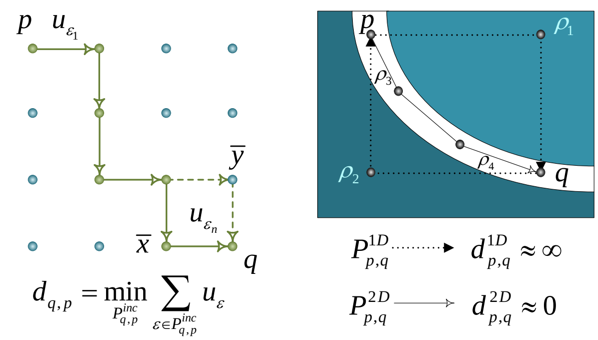


Fig. 1. Illustration of (left) an incremental path, (right) difference between 1D and 2D geodesic distances and its corresponding paths. The graph vertex ρ_1 or alternative ρ_2 is an intermediate node of the path $P^{1D}_{p,q}$, and vertices ρ_3, ρ_4 are intermediate nodes for the path $P^{2D}_{p,q}$. In this illustration we assume that the distance between the intensities I_p and I_{ρ_1} to be ∞, then $d^{1D}_{p,q} \approx \infty$. In contrast the path $P^{2D}_{p,q}$ can be chosen in such a way via vertices ρ_3 and ρ_4 that $d^{2D}_{p,q} \approx 0$, here we assume that the spacial term $\delta = 0$.

A. Geodesic distance approximation

We use $p, q, k, l \in V$ to identify vertices, and set $\varepsilon = (k, l) \in E$ as edges of an image graph $G = \{V, E\}$. The geodesic distance based filter is usually chosen in the following form, which makes the filter recursive

$$\begin{aligned} F_q &= \tfrac{1}{W_q} \sum_{p \in V} e^{-a d_{p,q}} f_p, \\ W_q &= \sum_{p \in V} e^{-a d_{p,q}}, \end{aligned} \tag{1}$$

where F_q and f_p are the output and the input of the filter respectively. A weight $e^{-a d_{p,q}}$ in (1) defines an affinity between any two image pixels (p, q) (further replaced by a shortcut $w_{p,q} = e^{-a d_{p,q}}$) and W_q is the normalization factor. The variable $d_{p,q}$ in (1) is the geodesic distance between image pixels (p, q) which for an image I_p can be defined on the discrete grid graph as

$$\begin{aligned} d_{p,q} &= \min_{P_{p,q}} \sum_{\varepsilon \in P_{p,q}} u_\varepsilon, \\ u_{\varepsilon=(k,l)} &= \|I_k - I_l\| + \delta, \end{aligned} \tag{2}$$

where $P_{p,q}$ is any path between two graph vertices (p, q) and δ is the spacial term. The norm in (2) is the Euclidean distance in RGB space. If one considers a grayscale image the norm reduces to the simple absolute value. Input function f_p in (1) can be the same as an image I_p, but in many applications the image I_p only defines the affinity space of the filter.

Note, the filter intrinsic parameters a and δ in (2) approximately correspond to the parameters of the classic bilateral filter with the Gaussian kernel as follows

$$a = \frac{2}{\sigma_r^2}, \delta = \frac{\sigma_r^2}{\sigma_s^2}, \tag{3}$$

where σ_r^2 and σ_s^2 are the range and the spacial variance respectively.

A path $P_{p,q}$ ideally should minimize the sum in (2) to be the geodesic distance. However the recursive algorithms proposed in [12], [29] realize only two consecutive 1D recursion: first along X direction then along Y (or vice versa). The limitation of this approximation is illustrated in Fig. 1(right) by the $P^{1D}_{p,q}$ path. In this illustration we assume that the distance between the intensities I_p and I_{ρ_1} to be ∞ then $d^{1D}_{p,q} \approx \infty$. The infinity distance means that there is no influence or connectivity between nodes p and q. As a consequence, filters based on this estimation of the geodesic distance, have a smaller support region, and are therefore of lower quality. We use the term support region to indicate the set of pixels from a neighborhood with non-negligible influence or weights of the filter kernel.

Our algorithm aims to improve the geodesic distance estimation by approximating the optimal incremental distance, which is illustrated in Fig. 1(left). The incremental path $P^{inc}_{p,q}$ is characterized by L1 Euclidean distance between initial pixel p and any next pixel, which belongs to the path. The value of this L1 distance monotonically increases along the path. Note that the $P^{inc}_{p,q}$ path is more flexible than the fixed path $P^{1D}_{p,q}$, but still constrained in comparison with an arbitrary path $P_{p,q}$ used in the general geodesic distance definition Eq. (2). In turn the path in Fig. 1(left) can be estimated recursively by a dynamic programming algorithm only choosing the minimum between two incoming edges plus the optimal path ending in the corresponding neighbor pixels $u_{\bar{x},q} + d_{p,\bar{x}}$ and $u_{\bar{y},q} + d_{p,\bar{y}}$ of any final pixel q of the path (here two shortcuts $\bar{x} = \{x_q - 1, y_q\}$ and $\bar{y} = \{x_q, y_q - 1\}$ are used). This fact inspired us to include the dynamic programming principle into our improved geodesic distance computation. The dynamic programming principle provides the exact solution when considering two points (one starting and one ending point). However, it cannot be applied exactly for the recursive weight calculation between sets of points, which is necessary for geodesic filtering. Nevertheless, the dynamic programming principle can be considered a motivation for our approach to estimating the geodesic distance.

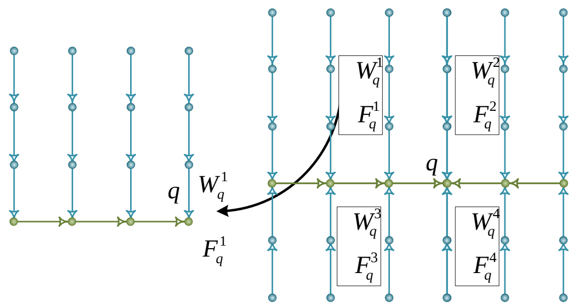


Fig. 2. Illustration of the general recursive summation tree scheme for the 1D recursion first along Y and then along X and its part of the first quadrant.

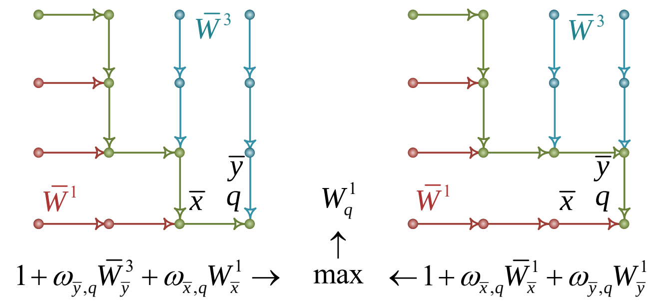


Fig. 4. Illustration of the proposed recursive summation tree growing process for the first quadrant.

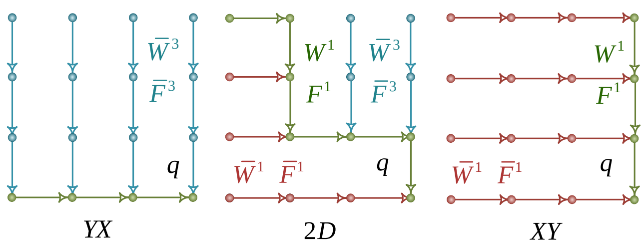


Fig. 3. Illustration of all three possible realization of recursive summation trees: (left) - the tree corresponds to the scheme in Eq. (11) first along Y and then along X direction; (right) - s the tree corresponds to the scheme in Eq. (12) first along X and then along Y direction; (middle) - the tree corresponds to the proposed 2D scheme Fig. 4.

From (2) one can derive

$$w_{p,q} = e^{-a \min\limits_{P^{inc}_{q,p}} \sum\limits_{\varepsilon \in P^{inc}_{p,q}} u_\varepsilon} = \max_{P^{inc}_{p,q}} \prod_{\varepsilon \in P^{inc}_{p,q}} e^{-a u_\varepsilon} = \max_{P^{inc}_{p,q}} \prod_{\varepsilon \in P^{inc}_{p,q}} \omega_\varepsilon \quad (4)$$

Eq. (4) states that the distance minimization is equivalent to maximization of the weight $w_{p,q}$. The weight $w_{p,q}$ itself is a measure of a reciprocal influence between two pixels (p, q). From Fig. 1(right) we can see that $d^{2\mathrm{D}}_{p,q} \approx 0$ whereas $d^{1\mathrm{D}}_{p,q} \approx \infty$. Consequently, the affinity weights are $w^{2\mathrm{D}}_{p,q} \approx 1$ and $w^{1\mathrm{D}}_{p,q} \approx 0$. It means that for the 2D path there is a strong influence between the pixel q and the pixel p and no influence in the case of the 1D path. The main idea is to maximize the integral weight factor W_p or to minimize the sum of all distances, which end at the pixel q.

We will further explain our algorithm on the basis of recursive calculation trees, which are depicted in Fig. 2 - 4. To formalize this tree growing process let us map the graph G into the 2D regular image grid $\{X, Y\}$: $p = \{x_p, y_p\}$. Then the four connected graph edges are $\varepsilon = (k, l) \Rightarrow x_k = x_k \pm 1, y_k = y_l$ or $x_k = x_l, y_k = y_l \pm 1$. The full calculation tree of Eq. (1) can be divided into four quadrants (see the four branches of the calculation tree in Fig. 2) and in these quadrants four base sums and their integral weights can be calculated

$$F^i_q = \begin{cases} F^1_q = \sum\limits_{y_p \le y_q} \sum\limits_{x_p \le x_q} w_{p,q} f_p \\ F^2_q = \sum\limits_{y_p \le y_q} \sum\limits_{x_p \ge x_q} w_{p,q} f_p \\ F^3_q = \sum\limits_{y_p \ge y_q} \sum\limits_{x_p \le x_q} w_{p,q} f_p \\ F^4_q = \sum\limits_{y_p \ge y_q} \sum\limits_{x_p \ge x_q} w_{p,q} f_p \end{cases} \quad (5)$$

$$W^i_q = \begin{cases} W^1_q = \sum\limits_{y_p \le y_q} \sum\limits_{x_p \le x_q} w_{p,q} \\ W^2_q = \sum\limits_{y_p \le y_q} \sum\limits_{x_p \ge x_q} w_{p,q} \\ W^3_q = \sum\limits_{y_p \ge y_q} \sum\limits_{x_p \le x_q} w_{p,q} \\ W^4_q = \sum\limits_{y_p \ge y_q} \sum\limits_{x_p \ge x_q} w_{p,q} \end{cases} \quad (6)$$

Note, functions F^i_q and W^i_q are the result of summation over all the pixels p in the corresponding quadrant domain defined in (5-6). However the weights $w_{p,q}$ in this summation formula correspond to paths conditioned by one of the three recursive calculation trees illustrated in Fig. 3. In other words any path must be along the edge direction (arrows Fig. 3) in the directed graph trees Fig. 3. So, rather than only considering the vertical and horizontal recursion we introduce an additional recursion tree to improve the estimation of the geodesic distance. This recursion tree differs from the horizontal and vertical recursion trees because it adapts to the image content. The growing process will be further explained in the following Subsection.

We also introduce four support sums and their corresponding integral weights, which are important component of our algorithm. Note, the following sums are the 1D summation along image rows or columns:

$$\bar{F}^i_q = \begin{cases} \bar{F}^1_q = \sum\limits_{x_p \le x_q; y_p = y_q} w_{p,q} f_p \\ \bar{F}^2_q = \sum\limits_{x_p \ge x_q; y_p = y_q} w_{p,q} f_p \\ \bar{F}^3_q = \sum\limits_{x_p = x_q; y_p \le y_q} w_{p,q} f_p \\ \bar{F}^4_q = \sum\limits_{x_p = x_q; y_p \ge y_q} w_{p,q} f_p \end{cases} \quad (7)$$

$$\bar{W}_q^i = \begin{cases} \bar{W}_q^1 = \sum\limits_{x_p \le x_q; y_p = y_q} w_{p,q} \\ \bar{W}_q^2 = \sum\limits_{x_p \ge x_q; y_p = y_q} w_{p,q} \\ \bar{W}_q^3 = \sum\limits_{x_p = x_q; y_p \le y_q} w_{p,q} \\ \bar{W}_q^4 = \sum\limits_{x_p = x_q; y_p \ge y_q} w_{p,q} \end{cases} \tag{8}$$

Finally, Eq. (1) can be decomposed into the next expression:

$$\begin{array}{c} F_q = \frac{1}{W_q} \left(\sum\limits_{1 \le i \le 4} F_q^i - \sum\limits_{1 \le i \le 4} \bar{F}_q^i + f_q \right) \\ W_q = \sum\limits_{1 \le i \le 4} W_q^i - \sum\limits_{1 \le i \le 4} \bar{W}_q^i + 1 \end{array} \tag{9}$$

Here Eq. (9) assumes that the sum of four functions F_q^i counts each support function $\bar{F}_q^i$ twice due to domains overlap. Thus we need to subtract the sum $\sum\limits_{1 \le i \le 4} \bar{F}_q^i$ to be consistent with Eq. (1).

B. Recursive implementation

All four base sums from (5-6) and support sums from (7-8) can be calculated recursively. Support sums should be calculated first according to:

$$\begin{array}{l} \bar{F}_q^i = f_q + \omega_{\bar{q}_i,q} \bar{F}_{\bar{q}_i}^i \\ \bar{W}_q^i = 1 + \omega_{\bar{q}_i,q} \bar{W}_{\bar{q}_i}^i \\ \bar{q}_1 = \{x_q - 1, y_q\} \\ \bar{q}_2 = \{x_q + 1, y_q\} \\ \bar{q}_3 = \{x_q, y_q - 1\} \\ \bar{q}_4 = \{x_q, y_q + 1\} \end{array} \tag{10}$$

In the same manner the value of F_q^i and W_q^i can be calculated recursively:

$$\begin{array}{l} F_q^i = f_q + \omega_{\bar{q}_i,q} \bar{F}_{\bar{q}_i}^{\bar{i}} + \omega_{q_i,q} F_{q_i}^i \\ W_q^i = 1 + \omega_{\bar{q}_i,q} \bar{W}_{\bar{q}_i}^{\bar{i}} + \omega_{q_i,q} W_{q_i}^i \\ q_1 = q_2 = \{x_q, y_q - 1\} \\ q_3 = q_4 = \{x_q, y_q + 1\} \\ \bar{q}_1 = \bar{q}_3 = \{x_q - 1, y_q\} \\ \bar{q}_2 = \bar{q}_4 = \{x_q + 1, y_q\} \\ \bar{i} = \begin{cases} 1 | i \in \{1, 3\} \\ 2 | i \in \{2, 4\} \end{cases} \end{array} \tag{11}$$

This calculation scheme corresponds to the 1D recursion in [12], [29] first along X direction and then along Y. One branch of this particular calculation tree is illustrated in Fig. 3 (right). And for another order first Y and then X direction the same values can be alternatively calculated

$$\begin{array}{l} F_q^i = f_q + \omega_{\bar{q}_i,q} \bar{F}_{\bar{q}_i}^{\bar{i}} + \omega_{q_i,q} F_{q_i}^i \\ W_q^i = 1 + \omega_{\bar{q}_i,q} \bar{W}_{\bar{q}_i}^{\bar{i}} + \omega_{q_i,q} W_{q_i}^i \\ q_1 = q_3 = \{x_q - 1, y_q\} \\ q_2 = q_4 = \{x_q + 1, y_q\} \\ \bar{q}_1 = \bar{q}_2 = \{x_q, y_q - 1\} \\ \bar{q}_3 = \bar{q}_4 = \{x_q, y_q + 1\} \\ \bar{i} = \begin{cases} 3 | i \in \{1, 2\} \\ 4 | i \in \{3, 4\} \end{cases} \end{array} \tag{12}$$

The first quadrant branch of this particular calculation tree is illustrated in Fig. 3 (left).

It can be seen, that in each point q there are two alternatives from which one can calculate F_q^1 and W_q^1, namely Eq. (11) or Eq. (12). Because our strategy is to maximize the integral influence at the pixel q, first we calculate W_q^1 by both Eq. (11) and Eq. (12) and then fix the choice based on which Eq. (11) or (12) gives the maximum value of W_q^1. Finally, our algorithm is growing a branch of a calculation tree, which is depicted in Fig. 3 (middle) and explained graphically in Fig. 4.

The proposed calculation scheme preserves the true incremental geodesic distance between pixels p and q if the pixel p is the only signal source (suppose we mask all unit weights except in the pixel p). We prove this assertion in the Appendix. Also in the Appendix we prove that the proposed calculation scheme Fig. 4 maximizes the value of W_q^1 relative to the direct implementation of the algorithm in (11) or (12), which are given in Fig. 3 (left and right respectively).

C. Illustration of improved geodesic distance

We perform three model experiments to show the difference between the proposed 2D and known 1D algorithms. Also, in this subsection we compare recursive algorithms with the algorithm based on the true geodesic distance calculation. Note, the true algorithm has demanding computational complexity: $O\left(|V|^2\right)$, which is excessive for most practical applications.

The first experiment shows the approximation accuracy of integral weights W_q for different recursive algorithms. Integral weights form the filter kernel and a low value of W_q mean that the pixel q is isolated in the geodesic distance affinity space and the filter kernel in this pixel is restricted to the pixel itself. Thus the false restriction due to wrong weights calculation leads to incorrect filter performance. In the appendix we prove that

$$W_q^{GT} \le W_q^{2D} \le \max\left(W_q^{XY}, W_q^{YX}\right), \tag{13}$$

where W_q^{GT} is the weight obtained by the true algorithm, W_q^{2D} is the weight calculated by the proposed method and W_q^{XY}, W_q^{YX} are resultant weights of two possible realization of the 1D algorithm.

To illustrate the impact of Eq (13) we show the approximation accuracy based on the weights calculation for a test image. We use the Lena color test image and calculate weights with kernel parameters: $\sigma_r = \sigma_s = 20$. The result is illustrated in Fig. 5. Here the weights $\max\left(W_q^{XY}, W_q^{YX}\right)$ (Fig. 5 (c)) corresponds to the kernel of the algorithm proposed in [29]. The idea of this technique is to calculated both recursions W_q^{XY} and W_q^{YX}, then in each pixel q choose the maximal value of the weights and the corresponding functions from pair: F_q^{XY} and F_q^{YX}. We can see that this simple weight composition also suffers from stripe structure as the initial 1D recursion (Fig. 5 (d)).

We also compare the results quantitatively on the base of PSNR criterion. For our algorithm approximation accuracy of W_q^{2D} to the ground truth W_q^{GT} is 29.91 dB. For the 1D recursion W_q^{XY} this approximation accuracy is 22.76 dB. And for the more accurate composite recursion $\max\left(W_q^{XY}, W_q^{YX}\right)$ approximation error is 24.46 dB, but it is still a considerably less accurate approximation.

The second experiment shows the filter response to the delta function signal in a homogeneous noisy image, and is illustrated in Fig. 6. In Fig. 6 (a) and (b) the 1D filter responses after one and four iterations respectively are shown. In Fig. 6 (c) and (d) are shown the 2D filter responses after one and two iterations respectively. Note, computational complexity of the 2D algorithm is twice as much as for the 1D algorithm. Thus the four iterations of 1D is equivalent to two iterations of the 2D filter. This experiment confirms the assumption that our algorithm is a better approximation of the 2D geodesic distance in the color space and its response is more robust to rotation. The experiment also confirms that the 1D algorithm is not rotation invariant.

The third experiment shows the filter response to the delta function signal in a narrow region of the test image and the result is illustrated in Fig. 7. In Fig. 7 (a) and (b) are shown the 1D filter responses after the first iteration and after four iterations respectively. In Fig. 7 (c) and (d) are shown the 2D filter responses after first iteration and after two iterations respectively. One can see that the region where the impact of the response is non zero is considerably greater in the case of the 2D realization of the filter than in the case of the 1D algorithm. Such an increasing of the kernel support region potentially increases accuracy in applications such as noise preserving filtering and stereo matching.

III. Denoising with geodesic distance based filter

As an example application of the proposed geodesic distance estimation we propose to look at image denoising. Direct application of the geodesic distance based filter (where the filtered image defines the affinity space $f_p \equiv I_p$) provides satisfactory denoising result especially for low level noise corruption due to the edge preserving ability of this filtering. However, the results are inferior compared to state-of-the-art denoising methods [6], [9], [16], [10], [11], [32], especially for high noise corruption level. This high noise sensitivity can be explained by the support region (filter kernel) restriction. In other words, high noise isolates many pixels of the filtered image, which form the neighborhood in the uncorrupted image. To address this phenomenon we propose to pre-filter the affinity space with the Gaussian kernel. More exactly, first we filter I

$$H = I * \mathcal{G}\left(0, \sigma_G^2\right), \tag{14}$$

where σ_G^2 is the Gaussian kernel variance. Then the function H is used for the filter affinity space formation in Eq. (2) and Eq. (4). Finally, this affinity space is used for the filter in Eq. (1), where f is a noisy image and F the reconstructed image. Note that the input image f remains unchanged and the filter output H in Eq. (14) is only applied for the affinity space computation. Here we aim to transform the filter affinity space to suppress the intrinsic outliers of this space caused by Gaussian noise. This local filter kernel reduction works only for the pixel-to-pixel affinity spaces like in the case of the classic bilateral filtering or the geodesic distance based filter. In contrast, filters that use patch affinity (such as non-local mean filter) cannot be improved by such a kernel transformation because this kernel initially includes local averaging due to the patch based norm. We found that the parameter σ_G (14) depends on the noise level and the noisy image gradient standard deviation. In our experiments this parameter is relatively small ($\sigma_G < 1.5$ for noise $\sigma_{noise} < 70$) and cannot smooth the edges even in the affinity space. In Section IV we motivate and explain more carefully how to calculate the σ_G parameter.

The proposed version of the recursive geodesic distance filters belongs to the class of filters that has computational complexity O(1). In other words, the total number of computational operations per image T_{img} is proportional to the total numbers of pixels in the image P: $T_{img} \propto \eta P$, where $\eta \ll P$. Because we calculate recursively 8 functions (F^i and $\bar{F}^i$) one might call our algorithm as O(8P) computational complexity. In contrast, most state-of-the-art approaches [6], [9], [10], [11], [16], [32] use nonlocal patch similarity and the complexity multiplier $\eta \gg 8$. Thus our filter is considerably faster. Here we have to note that one iteration of 1D recursive geodesic computation [12], [29] for a 2D image has O(4P) computational complexity and hence it is twice as fast as our algorithm. However, in [12], [29] multiple iterations of the algorithm are assumed (at least two iterations), thus our method obtains a more accurate approximation to the true geodesic kernel for the same computation time. On a single 2.4 GHtz CPU core, for processing a 1 megapixel color image the runtime is 0.32 seconds using two iterations of the geodesic distance filters 1D which is similar to the 0.29 seconds using one iteration of our approach.

For several test images the proposed filter outperforms state-of-the-art algorithms for denoising. In all cases we found that Gaussian filtering of the affinity space leads to improved results. In the experimental section we evaluate the geodesic distance filters both without (indicated by GDF) and with (indicated by GDF ($\sigma_G \neq 0$)) affinity space pre-filtering.

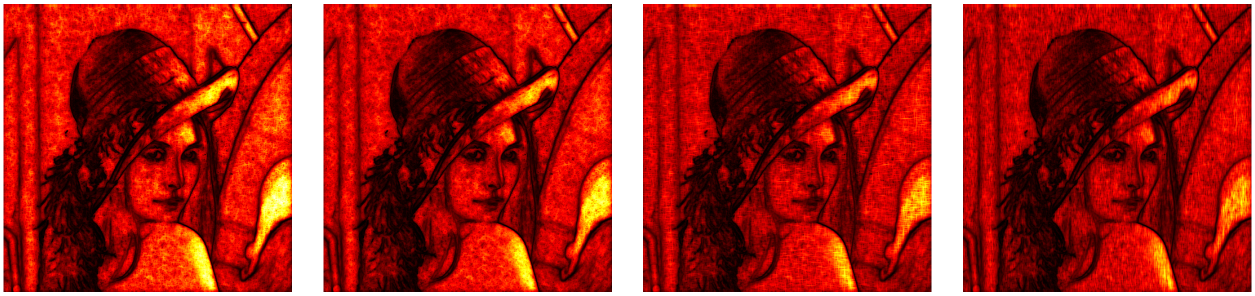
IV. Experiments

Our experimental section is divided into two parts. Firstly, we confirm the advantage of the proposed 2D geodesic distance filters realization over the known 1D realization implementing the geodesic distance filters to several experiments including edge preserving denoising. Then we consider the results of the proposed geodesic distance filters denoising filter in comparison with state-of-the-art methods.

A. Comparison two of 1D with 2D recursion

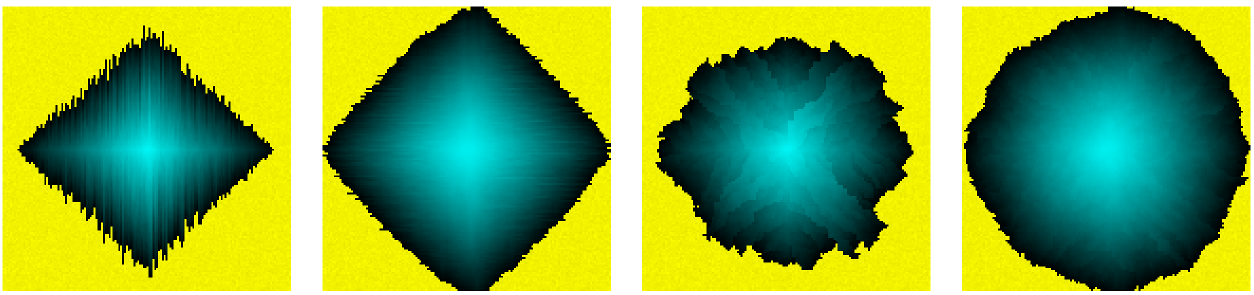
In Section II we show theoretically that our 2D recursion better approximates the true geodesic distance filters than the known 1D iterative recursions. In this Subsection we show that the proposed 2D recursive calculation also improves the result of the filter application.

The first experiment is the direct application of the geodesic distance filters (where the filtered image defines the affinity space $f_p \equiv I_p$) for the denoising problem. For this experiment we use the standard color test images including Parrots, Lena, Airplane, Peppers and Fruits images and put individual image comparisons in Table I.



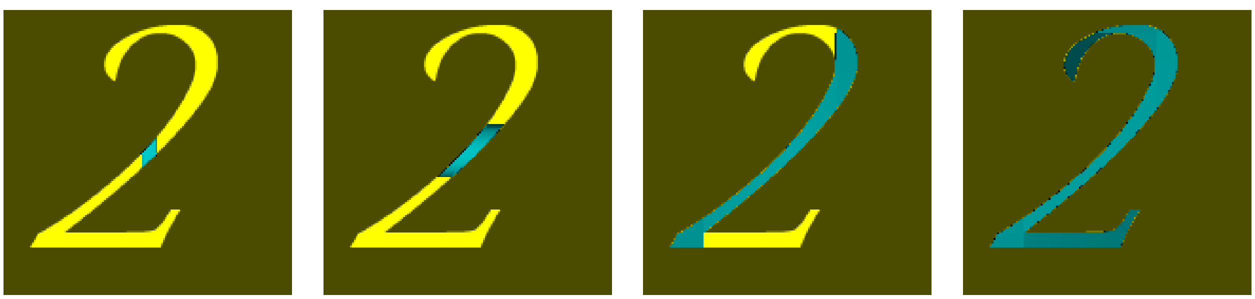
(a) (b) (c) (d)

Fig. 5. Illustration of the resultant weights W_q obtained by different true and recursive geodesic based kernels: (a) - the ground truth weights W_q^{GT}; (b) - the weights W_q^{2D} of the proposed approach; (c) - the composite weights $\max\left(W_q^{XY}, W_q^{YX}\right)$ [29]; (d) - the weights W_q^{XY} of 1D recursion.



(a) (b) (c) (d)

Fig. 6. Illustration of the filter response to the delta function signal in a homogeneous noisy image: (a) - the 1D filter responses after the first iteration; (b) - the 1D filter responses after four iterations; (c) - the 2D filter responses after the first iteration; (d) - the 2D filter responses after two iterations.



(a) (b) (c) (d)

Fig. 7. Illustration of the filter response to the delta function signal in a narrow region of the test image Two: (a) - the 1D filter responses after the first iteration; (b) - the 1D filter responses after four iterations; (c) - the 2D filter responses after the first iteration; (d) - the 2D filter responses after two iterations.

The filter parameters in Eq. (2) and Eq. (3) are chosen as follows

$$\sigma_r = \sigma_s = \alpha \sigma_{noise}, \tag{15}$$

where σ_{noise} is the Gaussian noise standard deviation and α is a parameter, which is equal to 1.3 for gray scale images and is equal to 2.4 ($\approx 1.3\sqrt{3}$) for RGB images. Here we assume that the standard deviation of our filter kernel is proportional to the standard deviation of the Gaussian noise, which is a usual assumption for denoising methods (e. g. the NLM algorithm). The parameter α is the averaged and rounded value of particular optimal values experimentally obtained using different noise levels ($3 < \sigma_{noise} < 70$) and different images (30 images of public internet datasets). Here the particular optimal value of the parameter supposes to maximize the value of the PSNR criteria for each individual noise and image. In our averaging experiments the difference between the proposed value and each individual PSNR maximum does not exceed 0.35 dB. Experimentally we found that the change of the α parameter by 5% modifies the PSNR inside the 0.3 dB interval.

Table I presents the PSNR (dB) values obtained by the proposed 2D and 1D recursive filters. The latter realization of the filter is presented in two versions: one and two iterations. In the case of two iterations, the kernel parameters must be consistent with the one iteration version. Thus we use the following formula which is taken from [12]:

$$\sigma_i = \sigma\sqrt{3}\frac{2^{N-i}}{\sqrt{4^N - 1}}, \tag{16}$$

where σ_i is the standard deviation for the kernel used in the i-th iteration, N is the total number of iterations, and σ is the standard deviation of the desired kernel. One can see in Table I that the proposed 2D filter outperforms its analogues realized by the 1D recursion.

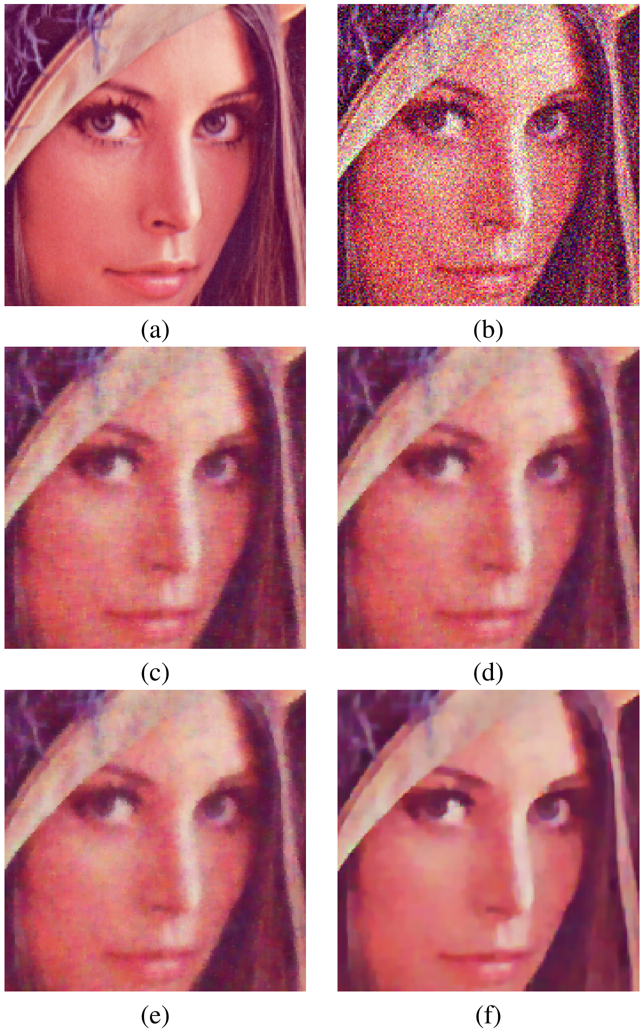


Fig. 8. Visual comparison of edge preserving denoising performance for the Lena test image with the 1D filter, the 2D filter and the proposed GDF ($\sigma_G \neq 0$) denoising filter: (a) - The zoomed original Lena test image; (b) - The original Lena test image contaminated by the Gaussian noise $\sigma_{noise} = 40$ (PSNR = 16.52 dB); (c) the result of filtering the Lena image contaminated by the Gaussian noise $\sigma_{noise} = 40$ with the 1D filter one iteration (PSNR = 26.62 dB); (d) - the result of filtering the Lena image with the 1D filter two iteration (PSNR = 26.83); (e)- the result of filtering with 2D filter (PSNR = 27.04 dB); (f) the result of filtering the Lena image with the proposed GDF ($\sigma_G = 1.25$) denoising filter (PSNR = 28.41).

Note, that even the 1D recursive filter is applied with two iterations, the quality of filtering is still worse than the filtering of the proposed 2D recursion. This fact is illustrated in Fig. 8, where a visual comparison shows that the filtering result of the proposed method is smoother than the conventional 1D approach. The experiment in Fig. 8 is performed with the Lena color test image contaminated by Gaussian noise $\sigma_{noise} = 40$ and we also include the result of the proposed GDF ($\sigma_G \neq 0$) denoising filter for visual comparison. The stripe like structure are most visible in Fig. 8 (b), where one iteration of the 1D recursion is applied.

To give a more solid proof to the claim in the paper we prepare Table II, where a comparison based on averaging over the Berkeley segmentation dataset [3] including 500 color images is proposed. We can see in Table II that the proposed 2D filter outperforms the realization with the 1D recursion.

In the second denoising experiment a test image with vertical stripes is taken and contaminated by Gaussian noise with the fixed value of sigma $\sigma_{noise} = 15$. Error values are measured in dB for different rotation angles β. The result of this experiment is summarized in Table III. One can see that the proposed approach is more robust to rotation.

B. Denoising experiments with Geodesic distance based filter

The second part of our experiment presents results of denoising in comparison with state-of-the-art methods: nonlocal means (NLM) [6], double noise similarity (DNS) [16], block-matching and 3D (BM3D) filtering [9], denoising with probabilistic patch-based weights (PPB) [10]. The experiments are carried out on standard test images: House, Cameraman, Lena, and Barbara, which are contaminated by additive white Gaussian noise with different variance σ^2_{noise}. The results of these experiments are summarized in Table IV, where the GDF method is the direct implementation of the geodesic distance filters with the intrinsic parameters $\sigma_s = \sigma_r$ as in Eq. (15). The proposed GDF ($\sigma_G \neq 0$) algorithm is described in Section III. The intrinsic parameters of the proposed method are:

$$\begin{array}{l} \sigma_G = \gamma \frac{\sqrt{2}\sigma_{noise}}{\sigma_{\Delta I}}, \\ \sigma_s = \sigma_r = \lambda + \mu\sigma_{noise}, \end{array} \tag{17}$$

where $\sigma_{\Delta I}$ is the standard deviation of the gradients of the noisy image I, and other constant factors are chosen as $\gamma = 1.2$, $\lambda = 3$ and $\mu = 0.3$. For the RGB images μ is equal to 0.5 ($\approx 0.3\sqrt{3}$). Note, in the scenario without noise our algorithm performs with nonzero parameters $\sigma_s = \sigma_r = 3$, however the impact of filtering in this case is almost neglectable because the order of the error is approximately equal to 50 dB. Here we again assume that the standard deviation of our affinity space transformation σ_G is proportional to the standard deviation of the Gaussian noise like in the case of the parameter α in (15). On the other hand this parameter should not over-smooth rich texture images, thus the parameter has to be inversely proportional to the gradient characteristics of the image. Because the affinity space transform in (14) compresses distances of this space non-linearly, the base parameter of the geodesic distance based filter kernel σ_s and σ_r is not simply proportional to the noise standard deviation, but depends on σ_{noise} linearly. Also the value of μ is now less than its analogues α (15). All three parameters in (17) are the averaged and rounded values of particular optimal values experimentally obtained using different noise levels and images, like in the case of the parameter estimation for Eq. (15). Parameters and performance of other filters in Table IV are taken from the paper [16].

From Table IV one can see that our GDF ($\sigma_G \neq 0$) algorithm outperforms state-of-the-art methods for the piecewise-constant images like House, Cameraman and shows comparable results for Lena and Barbara images. One can see that the proposed affinity space pre-filtering process considerably improves performance of the geodesic distance filter. It is interesting to note that our affinity space transform strategy improves the results of the classic bilateral filter and we

TABLE I
COMPARISON OF EDGE PRESERVING DENOISING PERFORMANCE FOR 1D FILTER AND 2D FILTER WITH DIFFERENT TEST IMAGES AND DIFFERENT GAUSSIAN NOISE σ_{noise}. ERROR VALUES ARE MEASURED IN dB

Algorithm	Parrots $\sigma_{noise} = 10$	Lena $\sigma_{noise} = 20$	Airplane $\sigma_{noise} = 30$	Pappers $\sigma_{noise} = 40$	Fruits $\sigma_{noise} = 50$
1D filter 1 iteration	35.08	29.72	27.58	25.28	24.04
1D filter 2 iterations	35.11	29.78	27.70	25.41	24.24
2D filter	35.52	30.21	28.16	25.96	24.92

TABLE II
COMPARISON OF EDGE PRESERVING DENOISING PERFORMANCE FOR 1D FILTER AND 2D FILTER BASED ON AVERAGING OVER BERKELEY SEGMENTATION DATASET OF 500 IMAGES FOR DIFFERENT GAUSSIAN NOISE σ_{noise}. ERRORS REPRESENT THE AVERAGE PSNR VALUES OVER ALL 500 IMAGES OF THE DATASET AND ARE MEASURED IN dB

Algorithm	$\sigma_{noise} = 20$	$\sigma_{noise} = 30$	$\sigma_{noise} = 40$	$\sigma_{noise} = 50$	$\sigma_{noise} = 60$	$\sigma_{noise} = 70$	$\sigma_{noise} = 80$	$\sigma_{noise} = 80$
1D filter 1 iteration	28.71	26.21	24.51	23.19	22.14	21.29	20.24	19.93
1D filter 2 iterations	28.78	26.34	24.63	23.33	22.29	21.46	20.75	20.15
2D filter	28.96	26.61	25.01	23.75	22.77	21.97	21.25	20.65

TABLE III
COMPARISON OF EDGE PRESERVING DENOISING PERFORMANCE FOR 1D FILTER AND 2D FILTER WITH A VERTICAL STRIPES TEST IMAGE AND DIFFERENT ROTATION ANGLES β. GAUSSIAN NOISE IN THIS EXPERIMENT IS FIXED $\sigma_{noise} = 15$. THE DISTANCE BETWEEN WHITE AND BLACK STRIPES IN THE EXPERIMENT IS 4 PIXELS. ERROR VALUES ARE MEASURED IN dB.

Algorithm	$\beta = 0°$	$\beta = 15°$	$\beta = 30°$	$\beta = 45°$	$\beta = 60°$	$\beta = 75°$	$\beta = 90°$
1D filter 1 iteration	32.7	32.3	32.1	32.1	32.2	32.4	32.8
1D filter 2 iterations	32.9	32.5	32.3	32.3	32.3	32.6	33.0
2D filter	34.8	34.7	34.8	34.8	34.8	34.7	34.7

include experiments with this filter (BF) in Table IV for both version: $\sigma_G = 0$ and $\sigma_G \neq 0$. For the bilateral filtering we use the same approach as is described in Eq. (14), however in this case we set the parameter $\alpha = 1.4$. (15) and the parameter $\gamma = 1.4$. (17). Also the results of filtering with the 1D recursive geodesic calculation (GDF 1D) are included in our final table for comparison.

In Fig. 9 qualitative performance of the proposed algorithm is illustrated. In these experiments we use the same four images as in previous experiments, but for different noise standard deviation: Barbara $\sigma_{noise} = 10$; Lena $\sigma_{noise} = 20$; Cameraman $\sigma_{noise} = 40$; House $\sigma_{noise} = 60$.

Almost all state-of-the-art method use the nonlocal patch similarity approach [6], [10], [16] or its modified versions [9], [11], [32]. Formally all these algorithm have O(1) = O(P) computational complexity, where P is the number of pixels, or O(PlogP) in the case where the fast Fourier transform is used. However the size of sliding windows that is used in this technique is usually comparable with the size of the image P. For example, the computation complexity of the NLM filter is O(21 × 21 × 7 × 7P). Thus this kind of algorithms is demanding. In contrast the proposed version of the recursive geodesic distance filters has computational complexity equal to O(8P). For the implementation of the Gaussian filters we also use the O(1) fast recursive filter [14], [30]. Thus our filter is considerably faster than most state-of-the-art approaches.

V. CONCLUSIONS

The main contribution of the presented paper is the 2D recursion, which improved performance of the conventional 1D geodesic distance filters. We show that our approach approximates the true geodesic distance filters better than the 1D geodesic distance filters and outperforms the result of the 1D filtering for denoising. The second contribution of the paper is a new denoising method. The idea is to pre-filter by the Gaussian filter to improve the affinity space for the further filtering with the geodesic distance based kernel. Experimental evaluation of our denoising method demonstrate comparable and for piecewise-constant images better results than state-of-the-art approaches, while our algorithm is faster with computational complexity O(8P).

In a future work we plan to extend application of the proposed filter. Our geodesic distance based recursive filter can be used in energy minimization problems with fully connected MRF model. Now for this purpose the bilateral filter with classical kernel is used and that approach has several theoretical drawbacks, which we intend to overcome with the proposed recursive geodesic distance filters. In addition, we are interested in applying the filter for optical flow, stereo and fast superpixel computation.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish project TIN2015-65464-R, TIN2016-79717-R and TIN2013-41751-P (MINECO/FEDER) and the COST Action IC1307 iV&L Net (European Network on Integrating Vision and Language), supported by COST (European Cooperation in Science and Technology) and the Catalan project 2014 SGR 221. .We also acknowledge the generous GPU support from Nvidia.

(Barbara) (Lena) (Cameraman) (House)

Fig. 9. Result of denoising with the GDF ($\sigma_G \neq 0$), the BF ($\sigma_G \neq 0$) and the NLM algorithms for different noise and different images: Barbara $\sigma_{noise} = 10$ ($\sigma_G = 0.68$); Lena $\sigma_{noise} = 20$ ($\sigma_G = 1.14$); Cameraman $\sigma_{noise} = 40$ ($\sigma_G = 1.27$); House $\sigma_{noise} = 60$ ($\sigma_G = 1.34$). (First row) original images; (Second row) noisy images; (Third row) images restored with the GDF; (Fourth row) images restored with the BF; (Fifth row) images restored with the NLM.

TABLE IV
PSNR ERROR VALUES OF ESTIMATED IMAGES WITH DIFFERENT DENOISING METHODS FOR IMAGES CORRUPTED BY DIFFERENT NOISE STANDARD DEVIATION σ_{noise}. THE VALUES OF σ_G HERE ARE CALCULATED ACCORDING TO EQ. (17) AND WE PUT THESE VALUES IN THE BRACKETS

Method	NLM [6]	PPB [10]	DNS2 [16]	BM3D [9]	BF	GDF 1D	GDF	BF ($\sigma_G \neq 0$)	GDF ($\sigma_G \neq 0$)
				House					
$\sigma = 10$	35.08	34.39	34.69	36.71	32.18	33.76	34.63	37.75 (1.21)	**38.00** (1.12)
$\sigma = 20$	32.38	31.70	32.08	33.77	26.37	29.34	30.89	**35.15** (1.29)	34.84 (1.19)
$\sigma = 40$	28.37	29.06	29.42	30.65	20.32	26.03	27.10	31.12 (1.36)	**31.19** (1.26)
$\sigma = 60$	25.69	26.62	27.57	**28.74**	17.75	22.87	25.10	27.75 (1.45)	28.16 (1.34)
				Cameraman					
$\sigma = 10$	30.78	30.51	31.52	34.18	31.92	33.24	33.70	34.15 (1.07)	**35.92** (0.99)
$\sigma = 20$	28.71	28.40	28.60	30.48	26.78	28.88	29.80	31.31 (1.28)	**32.39** (1.18)
$\sigma = 40$	25.91	25.74	26.23	27.18	20.63	24.29	25.67	27.98 (1.38)	**28.31** (1.27)
$\sigma = 60$	23.63	23.37	24.70	**25.32**	17.28	21.98	23.31	25.25 (1.46)	25.28 (1.34)
				Lena					
$\sigma = 10$	34.66	34.06	34.42	**35.93**	30.64	32.15	32.64	32.48 (1.04)	33.47 (0.96)
$\sigma = 20$	31.63	31.44	31.77	**33.05**	25.03	28.40	29.39	30.47 (1.24)	31.04 (1.14)
$\sigma = 40$	28.23	28.61	29.12	**29.86**	19.41	24.44	26.07	27.95 (1.35)	28.20 (1.25)
$\sigma = 60$	26.19	26.57	27.54	**28.27**	17.18	22.52	24.17	25.59 (1.44)	25.80 (1.33)
				Barbara					
$\sigma = 10$	33.25	32.09	32.56	**34.98**	29.80	30.78	30.96	30.13 (0.73)	31.46 (0.68)
$\sigma = 20$	30.32	29.35	29.79	**31.78**	24.08	26.35	26.71	25.40 (1.06)	26.01 (1.06)
$\sigma = 40$	26.42	26.84	26.99	**27.99**	18.94	22.76	23.56	23.74 (1.28)	23.83 (1.18)
$\sigma = 60$	24.13	24.57	25.07	**26.28**	16.42	21.16	22.16	22.56 (1.40)	22.67 (1.29)

APPENDIX

Proposition 1. The proposed calculation scheme in Fig. 4 preserve the true incremental geodesic distance between pixels if the pixel p is the only signal source (suppose we mask all unit weights except in the pixel p).

In other words, we need to prove that using alternative choice between Eq. (11) and Eq. (12) in calculation integral weights the following expression holds

$$d_{p,q} = \min_{P_{p,q}^{inc}} \sum_{\varepsilon \in P_{p,q}^{inc}} u_\varepsilon, \tag{18}$$

or taking into account Eq. (4) the equivalent expression must hold:

$$w_{p,q} = \max_{P_{p,q}^{inc}} \prod_{\varepsilon \in P_{p,q}^{inc}} \omega_\varepsilon \tag{19}$$

Firstly, let us exclude two trivial cases when the pixels p and q belong to the same vertical or horizontal grid line. In these particular cases Eq. (19) holds by our definition of the incremental path. One can find the optimal (in sense of Eq. (19)) incremental path and value $w_{p,q}$, corresponding to this path by implementing step by step dynamic programming optimization in each pixel. This calculation scheme is illustrated in Fig. 1 (left) and also can be formalized by the following expression:

$$w_{p,q} = \max \begin{pmatrix} \omega_{\bar{y},q} w_{p,\bar{y}} \\ \omega_{\bar{x},q} w_{p,\bar{x}} \end{pmatrix} \tag{20}$$

Note, the illustration in Fig. 1 (left) corresponds to the first quadrant and further we provide proves only for this quadrant, however for other quadrants proves can be derived in the similar way. Thus, we have to prove that the switched recursion between Eq. (11) and Eq. (12) achieves the same result in the case when the pixel p is the only signal source. The last condition can be formalized as follows

$$w_{k,k} = \begin{Bmatrix} 1 \leftarrow k = p \\ 0 \leftarrow otherwise \end{Bmatrix}, \tag{21}$$

then the switched recursion of Eq. (11) and Eq. (12) for integral weights becomes

$$W_q^1 = \max \begin{pmatrix} w_{q,q} + \omega_{\bar{x},q} \bar{W}_{\bar{x}}^1 + \omega_{\bar{y},q} W_{\bar{y}}^1 \\ w_{q,q} + \omega_{\bar{y},q} \bar{W}_{\bar{y}}^3 + \omega_{\bar{x},q} W_{\bar{x}}^1 \end{pmatrix} \tag{22}$$

Taking into account that integral weights $\bar{W}_{\bar{x}}^1$ and $\bar{W}_{\bar{y}}^3$ include only zero weights $w_{k,k}$ and also equal to zero, Eq. (21) can be simplified first to

$$W_q^1 = \max \begin{pmatrix} \omega_{\bar{y},q} W_{\bar{y}}^1 \\ \omega_{\bar{x},q} W_{\bar{x}}^1 \end{pmatrix}, \tag{23}$$

and then

$$W_q^1 = \max \begin{pmatrix} \omega_{\bar{y},q} w_{p,\bar{y}} \\ \omega_{\bar{x},q} w_{p,\bar{x}} \end{pmatrix}, \tag{24}$$

because integral weights $W_{\bar{y}}^1$ and $W_{\bar{x}}^1$ include only one non-zero weight $w_{p,p}$.

Comparing Eq. (20) and Eq. (24) one can see that both step by step recursions find optimal incremental path from the graph node p to the node q, and $W_q = w_{p,q}$. Thus we prove our **Proposition 1.**

Proposition 2. The proposed calculation scheme maximizes the integral weight W_q value relative to the direct implementation of the algorithms with Eq. (11) or Eq. (12).

As in the case of Proposition 1 the proof is derived only for the first quadrant of the calculation tree.

Our second assertion can be formalized as follows

$$\begin{matrix} W_q^{1:2D} \geq W_q^{1:XY}; \\ W_q^{1:2D} \geq W_q^{1:YX}. \end{matrix} \tag{25}$$

Let us prove this proposition by induction.

Firstly let us rewrite Eq. (22) as

$$W_q^{1:2D} = \max\begin{pmatrix} 1 + \omega_{\bar{x},q}\bar{W}_{\bar{x}}^{1} + \omega_{\bar{y},q}W_{\bar{y}}^{1:2D} \\ 1 + \omega_{\bar{y},q}\bar{W}_{\bar{y}}^{3} + \omega_{\bar{x},q}W_{\bar{x}}^{1:2D} \end{pmatrix} \tag{26}$$

If q is the only node of the tree in Fig. 3 (middle), then the proposition in Eq. (25) holds.

Thus we have to prove that if similar conditions hold for previous step formalized as follows

$$\begin{pmatrix} W_{\bar{y}}^{1:2D} \geq W_{\bar{y}}^{1:XY} & W_{\bar{y}}^{1:2D} \geq W_{\bar{y}}^{1:YX} \\ W_{\bar{x}}^{1:2D} \geq W_{\bar{x}}^{1:XY} & W_{\bar{x}}^{1:2D} \geq W_{\bar{x}}^{1:YX} \end{pmatrix}, \tag{27}$$

then Eq. (25) holds for the next step, in other word, for any node q.

The proof is the two follow chain inequalities:

$$\begin{aligned} W_q^{1:2D} &= \max\begin{pmatrix} 1 + \omega_{\bar{x},q}\bar{W}_{\bar{x}}^{1} + \omega_{\bar{y},q}W_{\bar{y}}^{1:2D} \\ 1 + \omega_{\bar{y},q}\bar{W}_{\bar{y}}^{3} + \omega_{\bar{x},q}W_{\bar{x}}^{1:2D} \end{pmatrix} \\ &\geq 1 + \omega_{\bar{x},q}\bar{W}_{\bar{x}}^{1} + \omega_{\bar{y},q}W_{\bar{y}}^{1:YX} = W_q^{1:YX}; \end{aligned} \tag{28}$$

and

$$\begin{aligned} W_q^{1:2D} &= \max\begin{pmatrix} 1 + \omega_{\bar{x},q}\bar{W}_{\bar{x}}^{1} + \omega_{\bar{y},q}W_{\bar{y}}^{1:2D} \\ 1 + \omega_{\bar{y},q}\bar{W}_{\bar{y}}^{3} + \omega_{\bar{x},q}W_{\bar{x}}^{1:2D} \end{pmatrix} \\ &\geq 1 + \omega_{\bar{y},q}\bar{W}_{\bar{y}}^{3} + \omega_{\bar{x},q}W_{\bar{x}}^{1:XY} = W_q^{1:XY}; \end{aligned} \tag{29}$$

which can be obtained by substitution condition from Eq. (27) to Eqs. (28)-(29).

References

[1] A. Adams, J. Baek, and M. Davis. Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum*, 29(2):753–762, 2010.

[2] X. An and F. Pellacini. AppProp: all-pairs appearance-space edit propagation. *ACM Transactions on Graphics*, 27(3):40, 2008.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.

[4] V. Aurich and J. Weule. Non-linear gaussian filters performing edge preserving diffusion. In *Mustererkennung 1995*, pages 538–545. Springer, 1995.

[5] E. Bennett, J. Mason, and L. McMillan. Multispectral bilateral video fusion. *IEEE Trans. Image Processing*, 16(5):1185–1194, 2007.

[6] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.

[7] A. Criminisi, T. Sharp, and A. Blake. Geos: Geodesic image segmentation. In *Computer Vision–ECCV 2008*, pages 99–112. Springer, 2008.

[8] A. Criminisi, T. Sharp, C. Rother, and P. Pérez. Geodesic image and video editing. *ACM Trans. Graph.*, 29(5):134, 2010.

[9] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[10] C.-A. Deledalle, L. Denis, and F. Tupin. Iterative weighted maximum likelihood denoising with probabilistic patch-based weights. *IEEE Transactions on Image Processing*, 18(12):2661–2672, 2009.

[11] W. Dong, X. Li, L. Zhang, and G. Shi. Sparsity-based image denoising via dictionary learning and structural clustering. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 457–464. IEEE, 2011.

[12] E. Gastal and M. Oliveira. Domain transform for edge-aware image and video processing. *ACM Transactions on Graphics*, 30(4):69, 2011.

[13] E. Gastal and M. Oliveira. Adaptive manifolds for real-time high-dimensional filtering. *ACM Transactions on Graphics*, 31(4):33, 2012.

[14] J.-M. Geusebroek, A. W. Smeulders, and J. Van De Weijer. Fast anisotropic gauss filtering. *Image Processing, IEEE Transactions on*, 12(8):938–943, 2003.

[15] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *European Conference on Computer Vision*, pages 725–739. Springer, 2014.

[16] G. Liu, H. Zhong, and L. Jiao. Comparing noisy patches for image denoising: A double noise similarity model. *IEEE Transactions on Image Processing*, 24(3):862–872, 2015.

[17] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang. On building an accurate stereo matching system on graphics hardware. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 467–474. IEEE, 2011.

[18] M. G. Mozerov. Constrained optical flow estimation as a matching problem. *Image Processing, IEEE Transactions on*, 22(5):2044–2055, 2013.

[19] M. G. Mozerov and J. van de Weijer. Accurate stereo matching by two-step energy minimization. *Image Processing, IEEE Transactions on*, 24(3):1153–1163, 2015.

[20] M. G. Mozerov and J. van de Weijer. Global color sparseness and a local statistics prior for fast bilateral filtering. *Image Processing, IEEE Transactions on*, 24(12):5842–5853, 2015.

[21] G. Papari, N. Idowu, and T. Varslot. Fast bilateral filtering for denoising large 3d images. *IEEE Transactions on Image Processing*, 26(1):251–261, 2017.

[22] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In *Proc. European Conf. on Computer Vision*, pages 568–580, 2006.

[23] R. Ramanath and W. E. Snyder. Adaptive demosaicking. *J. Electron. Imag.*, 12(4):633–642, 2003.

[24] S. M. Smith and J. M. Brady. Susana new approach to low level image processing. *International journal of computer vision*, 23(1):45–78, 1997.

[25] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[26] P. Wang, G. Zeng, R. Gan, J. Wang, and H. Zha. Structure-sensitive superpixels via geodesic distance. *International journal of computer vision*, 103(1):1–21, 2013.

[27] H. Winnemöller, S. Olsen, and B. Gooch. Real-time video abstraction. *ACM Transactions on Graphics*, 25(3):1221–1226, 2006.

[28] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *Proc. European Conf. on Computer Vision*, pages 211–224, 2006.

[29] Q. Yang. Recursive bilateral filtering. *Proc. European Conf. on Computer Vision*, pages 399–413, 2012.

[30] I. T. Young and L. J. Van Vliet. Recursive implementation of the gaussian filter. *Signal processing*, 44(2):139–151, 1995.

[31] K. Zhang, J. Lu, and G. Lafruit. Cross-based local stereo matching using orthogonal integral images. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(7):1073–1079, 2009.

[32] W. Zuo, L. Zhang, C. Song, and D. Zhang. Texture enhanced image denoising via gradient histogram preservation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1203–1210, 2013.

Mikhail G. Mozerov is a Senior Scientist at the Computer Vision Center Barcelona. He received his MS degree in physics from the Moscow State University, in 1982, and his PhD degree in digital image processing from the Institute of Information Transmission Problems, Moscow, in 1995. From 2006 to 2010, he was a Ramon y Cajal Fellow at the Universidad Autonoma de Barcelona. His research interests include signal and image processing, stereo and optical flow, and pattern recognition.

Joost van de Weijer is a Senior Scientist at the Computer Vision Center Barcelona and member of the LAMP team. He received the Ph. D. degree in 2005 from the University of Amsterdam. From 2005 to 2007, he was a Marie Curie Intra-European Fellow in the LEAR Team, INRIA Rhone-Alpes, France. From 2008 to 2012, he was a Ramon y Cajal Fellow at the Universidad Autonoma de Barcelona. His main research is on the usage of color information in computer vision application.