EDGE PRESERVING FILTERS USING GEODESIC DISTANCES ON WEIGHTED ORTHOGONAL DOMAINS

Luca Bertelli, B.S. Manjunath

Center for BioImage Informatics
Dept. of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106
{lbertelli, manj}@ece.ucsb.edu

ABSTRACT

We introduce a framework for image enhancement, which smooths images while preserving edge information. Domain (spatial) and range (feature) information are combined in one single measure in a principled way. This measure turns out to be the geodesic distance between pixels, calculated on weighted orthogonal domains. The weight function is computed to capture the underlying structure of the image manifold, but allowing at the same time to efficiently solve, using the Fast Marching algorithm on orthogonal domains, the eikonal equation to obtain the geodesic distances. We show promising results in edge-preserving denoising of gray scale, color and texture images.

***Index Terms*—** Adaptive smoothing filters, geodesic distance, Fast Marching Method, edge-preserving filtering.

1. INTRODUCTION

The goal of denoising by *filtering*, probably the most fundamental operation in image processing, is to remove noise from images while preserving the signal content. Low-pass filtering using Gaussian kernels, for instance, exploits the fact that for neighboring pixels signal components are highly correlated, while noise components tend to be uncorrelated. Hence computing weighted averages of pixel values in the neighborhood removes noise while preserving signal. This model does not take into account abrupt local variations of the image (such edges) and therefore is not suitable for application where edge-preserving smoothing is required.

One well established approach to exploit local information in the filtering process is anisotropic diffusion [1]. By solving a locally weighted diffusion equation, images are selectively smoothed. In other words pixels are averaged using space dependent kernels, whose size and weight coefficients are computed using local information. A somehow related approach was proposed in [2], where space depend binary kernels are used for anisotropic averaging (G-neighbors). In [3] bilateral filters are introduced as nonlinear filters which combine domain and range filtering. The convolution kernels are in fact products of two component: domain (which represent spatial closeness) and range (which represent similarity in the feature space).

In this paper we combine range and domain information using geodesic distances between pixels, evaluated on weighted orthogonal domains. The proposed algorithm is computationally efficient (distances are computed using Fast Marching on orthogonal domains, as opposed to [4] where triangulated domains are used) and independent on the number of image channels. In addition a relation is established between range and domain components via the concept of geodesic distance and cost of crossing pixels during distance calculation.

2. GEODESIC DISTANCES ON WEIGHTED DOMAINS AND EDGE PRESERVING FILTERS

Define an image I as the mapping :

$$\begin{aligned} I &: \mathbb{R}^2 \rightarrow \mathbb{R}^n \\ I &: (x,y) \rightarrow \left(I^1(x,y), I^2(x,y), \ldots, I^n(x,y)\right) \end{aligned} \tag{1}$$

where n is the number of channels. We can now introduce the matrix $\mathcal{G}$ defined as follows:

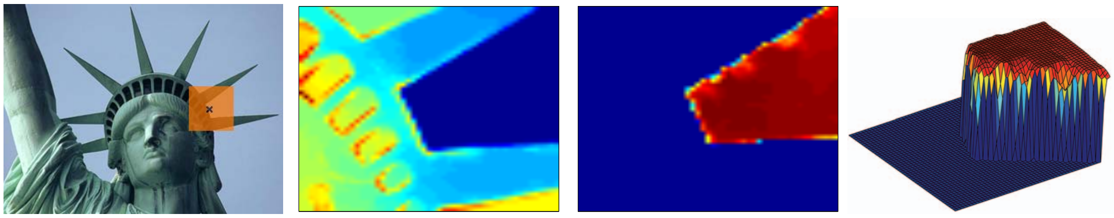
$$\mathcal{G} = (g_{ij}) = \begin{pmatrix} \sum_{i=1}^{n} \left(I_x^i\right)^2 & \sum_{i=1}^{n} I_x^i I_y^i \\ \sum_{i=1}^{n} I_x^i I_y^i & \sum_{i=1}^{n} \left(I_y^i\right)^2 \end{pmatrix} \tag{2}$$

where $I_x^i \triangleq \frac{\partial I^i}{\partial x}$ and $I_y^i \triangleq \frac{\partial I^i}{\partial y}$. The matrix $\mathcal{G}$ is positive semidefinite, since it is symmetric and all its principal minors are non negative. Hence both its eigenvalues λ_1 and λ_2 are non negative. These eigenvalues contain information about the multichannel gradients of the image I [5].

We propose to use the biggest eigenvalue $\lambda_1 \geq \lambda_2$ to weight an orthogonal domain and to compute distances on this domain. Distances on orthogonal domain can be computed by solving an *eikonal equation* with appropriate boundary conditions:

$$|\nabla T(x,y)| = \lambda_1(x,y) \tag{3}$$

This work was supported by the National Science Foundation award NSF ITR-0331697.



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

Fig. 1. (a) An image of the statue of liberty with highlighted in orange a window 51×51 center at pixel (120,238). (b) Geodesic distances from the center pixel to the rest of the pixels within the window. (c) Filter coefficients obtained according to (8) setting $\sigma = 1$. (d) 3D view of the filter window.

The eigenvalue $\lambda_1(x, y)$ gives the local weights used in arc-lenght calcualtion: $ds^2 = \lambda_1^2(x, y)(dx^2 + dy^2)$. By setting as boundary condition $T(x_0, y_0) = 0$ and solving for T gives distances from the point (x_0, y_0) to the point (x, y):

$$T(x, y) = \int_{(x_0,y_0)}^{(x,y)} ds \tag{4}$$

If the weight function is constant over the whole domain, the solution of the eikonal equation is exactly the Euclidean distance. In our case, since the weight function $\lambda_1(x, y)$ is proportional to the multichannel gradient, the distance between two pixels separated by high gradient will be higher that the distance between two pixels belonging to the same low gradient area.

The process of filtering an image $I(\mathbf{x})$ $(\mathbf{x} = (x, y))$ with an isotropic low pass filter can be described through the following convolution operation (if I is multichannel the convolution is intended channel by channel):

$$I_{out}(\mathbf{x}) = \frac{1}{c(\mathbf{x})} \int_{-\infty}^{+\infty} I(\mathbf{x}')\kappa(\mathbf{x}, \mathbf{x}')d\mathbf{x}' \tag{5}$$

where the normalization coefficient $c(\mathbf{x})$ is defined so that the DC component of the signal is preserved. Hence:

$$c(\mathbf{x}) = \int_{-\infty}^{+\infty} \kappa(\mathbf{x}, \mathbf{x}')d\mathbf{x}' \tag{6}$$

If $\kappa(\mathbf{x}, \mathbf{x}') = \kappa(\mathbf{x} - \mathbf{x}')$, the filter is shift invariant and the low-pass operation is performed in an isotropic way. The image is therefore smoothed isotropically, without preserving the edges.

In order to preserve the edges in the filtering process, we propose to define the kernel function κ as a function of the geodesic distance between two points on the domain as defined in (4) :

$$\kappa(\mathbf{x}, \mathbf{x}') = f(\int_{\mathbf{x}}^{\mathbf{x}'} ds) \tag{7}$$

For example using a Gaussian function of the geodesic distance between two points κ becomes:

$$\kappa(\mathbf{x}, \mathbf{x}') = e^{-\frac{1}{2}\left(\frac{\int_{\mathbf{x}}^{\mathbf{x}'} ds}{\sigma}\right)^2} \tag{8}$$

The eikonal equation in (3) has to be solved once for every pixel in the image in order to compute geodesic distances to the neighbors within the filter window. The Fast Marching Method [6] is an efficient algorithm to solve the eikonal equation with complexity $O(n \log n)$, where n is the number of points in the orthogonal grid. In our case we can restrict the orthogonal grid to a window of the size of the filter, centered around each pixel. The total complexity of the distance computation process is therefore $O(n_i \cdot n_w \log n_w)$, where n_i is the number of pixel in the image and n_w is the number of pixel in the filter support window. Assuming fixed filter size, the distance computation complexity grows therefore linearly with the image size.

3. COMPARISON WITH PREVIOUS WORK

3.1. Beltrami Flow

The Beltrami flow [4] originates from minimizing the area of a 2D Riemannian manifold embedded in $\mathbb{R}^{n+2}$ (where n is the number of channels of the image I). The points on the manifold are specified by $\left(x, y, I^1(x, y), \cdots, I^n(x, y)\right)$ and therefore the metric of the manifold is:

$$\mathcal{M} = (m_{ij}) = \begin{pmatrix} 1 + \sum_{i=1}^n \left(I_x^i\right)^2 & \sum_{i=1}^n I_x^i I_y^i \\ \sum_{i=1}^n I_x^i I_y^i & 1 + \sum_{i=1}^n \left(I_y^i\right)^2 \end{pmatrix} \tag{9}$$

Note that $m_{ij} = \delta_{ij} + g_{ij}$, where g_{ij} was defined in (2). The Beltrami flow is then derived as steepest descent minimization of the area of the manifold:

$$S = \iint \sqrt{\det(\mathcal{M})} dx dy \tag{10}$$

The PDE of the flow, as results of the minimization process is:

$$I_t^i = \frac{1}{\sqrt{\det(\mathcal{M})}} \mathrm{Div}\left(\sqrt{\det(\mathcal{M})}\mathcal{M}^{-1}\nabla I^i\right) \tag{11}$$

In [7] an iterative implementation of the PDE is replaced with a one step filter using a short time kernel. It turns out that the filter kernel can be written as:

$$\kappa_{BE}(\mathbf{x}, \mathbf{x}', t) = \frac{H}{t} e^{-\frac{1}{4}\frac{(\int_{\mathbf{x}}^{\mathbf{x}'} ds)^2}{t}} \tag{12}$$

where now $ds = m_{11}dx^2 + 2m_{12}dxdy + m_{22}dy^2$ is an arc-length element on the manifold and geodesic distances are now intended as distances on the manifold.

The main difference with respect to the method that we propose in this paper consists in the fact that the eikonal equation has to be solved on non orthogonal coordinate system (the manifold). This is due to the term $2m_{12}dxdy$ in the arc-length calculation. The presence of obtuse angles may occur and therefore a preprocessing stage is necessary to split every obtuse angle in two acute ones.

In our model we embed the information about the geometry of the manifold in the weight term $\lambda_1(x, y)$, avoiding therefore this problem. Geodesic distances are in fact evaluated on orthogonal weighted domains and the Fast Marching Method can be used in its original formulation.

3.2. Bilateral Filters

Bilateral Filtering [3] uses a filter kernel composed by two terms: the domain term captures information about spatial distances between pixels and the range term captures information about distances in the feature space. This kernel can be written as:

$$\kappa_{BI}(\mathbf{x}, \mathbf{x}') = e^{-\frac{1}{2}\left(\frac{||\mathbf{x}-\mathbf{x}'||}{\sigma_d}\right)^2} e^{-\frac{1}{2}\left(\frac{||I(\mathbf{x})-I(\mathbf{x}')||}{\sigma_r}\right)^2} \tag{13}$$

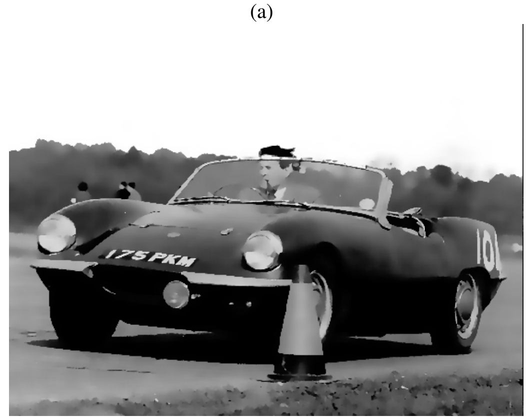
The main asset of this approach is that it does not rely on image derivatives. On the other hand the computation complexity is dependent on the number of channel of the image (i.e. the dimension of the feature space). With the approach proposed in this paper, we combine domain and range information in the weight function $\lambda_1(x, y)$ in a more principled way. In addition the geodesic distances computation is independent of the dimension of the feature space.

4. EXPERIMENTAL RESULTS

In this section we present an evaluation of the proposed algorithm on gray scale, color and texture images. In all the experiments a filter window 11×11 pixels is used and σ in equation (8) is set to 1.[1] Fig. 2 is meant to show the capability to denoise a gray scale image, while preserving edges. In the output image (Fig. 2 (b)), edges are still sharp and small scale details, as for example the face of the driver or the steering wheel, are preserved, while noise is removed.

Fig. 3 shows the potential of the proposed algorithm for removing texture from color images. In Fig. 3 (b) texture is removed from the cheek, the mouth and the ears of the puma,

(a)

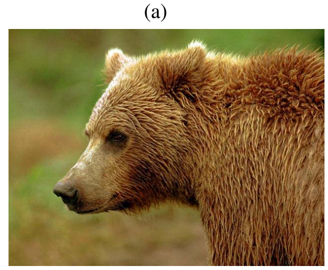


(b)

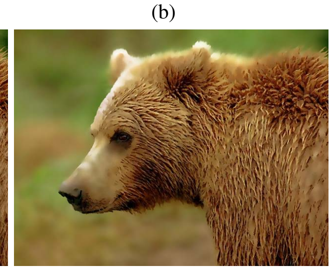
Fig. 2. (a) Noisy gray scale image. (b) Proposed algorithm output using the filter kernel (size 11×11) in (8) with $\sigma = 1$.

(a)

(b)



(c)



(d)

Fig. 3. (a,c) Original images (480×640). (b,d) Filtered images using the filter kernel (size 11×11) in (8) with $\sigma = 1$. Running time is ≈ 32 s on a P4 3 Ghz with 1 GB of RAM.

[1]The pictures of the experimental results can be seen full size at: http://vision.ece.ucsb.edu/∼lbertelli/icip/geodesic_filtering.

(a) (b) (c)

Fig. 4. Outputs of filtering the noisy image in Fig. 2(a) using the rescaled weight $\eta(x, y)$. (a) $k = 0$, (b) $k = 50$, (c) $k = 500$. In (c) fine scale details, as the steering wheel, are perfectly preserved, even better than in Fig. 2(a) where rescaling is not used.

while sharp edges are preserved, Notice that the whiskers, even if much smaller than the filter size, are still crisp after filtering. Similar considerations apply to Fig. 3 (c,d), where fine texture is removed from the snout and the forehead of the bear, while most of the details about the wet fur are preserved.

One interesting property of bilateral filters [3] is that, by changing the parameters σ_d and σ_r in (13), more importance can be given to range filtering than domain filtering or vice versa. These two parameters, which control the shapes of the two gaussian components of the filter kernel, can assume arbitrary values and it is difficult to relate them in a meaningful way. Within the framework proposed in this paper, we introduce the following rescale of the weight function $\lambda_1(x, y)$, so that range and domain filtering are meaningfully related through the concept of geodesic distance between pixels. Let us define the rescaled weight function:

$$\eta(x, y) = \frac{k}{\lambda_{MAX}} \lambda_1(x, y) + 1 \tag{14}$$

where $\lambda_{MAX} = \max \lambda_1(x, y)$. Since $\eta(x, y)$ is used to weigh the image domain in geodesic distance computation, can be thought as the cost of crossing each pixel. The higher the cost the faster distances will increase. k is chosen so that, if 1 is the cost of crossing a flat region ($\lambda(x, y) = 0$), $k + 1$ is the cost of crossing the strongest edge ($\lambda(x, y) = \lambda_{MAX}$). The parameter k acts as weight between domain and range contribution. If $k \ll 1$ then $\eta(x, y) \approx 1$ over the whole image and geodesic distances become euclidean distances. Fig. 4(a) shows the output of the filter (for the noisy image in Fig. 2(a)) using the rescaled weight function η and $k = 0$. The filter kernel is isotropic and the domain component is dominant. Increasing k (Fig. 4(b,c) with k=50 and k=500 respectively), crossing edges becomes more and more costly and therefore the weight of the range component is increased.

5. CONCLUSIONS

We introduced a computationally efficient and mathematically sound framework for smoothing images while preserving edges. Geodesic distances between pixels, efficiently computed on weighted orthogonal domains using the Fast Marching algorithm, are used as combination of domain and range information. A relationship between domain and range component is established using the concept of geodesic distance (and cost of crossing pixels), via a rescaling of the weight function. Experimental results in denoising and texture removal are shown.

6. REFERENCES

[1] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 629–639, July 1990.

[2] T. Boult, R. A. Melter, F. Skorina, and I. Stojmenovic, "G-neighbors," in *Proc. SPIE Conference on Vision Geometry II*, 1993, pp. 96–109.

[3] Carlo Tomasi and Roberto Manduchi, "Bilateral filtering for gray and color images," in *IEEE International Conf. on Computer Vision (ICCV)*, Bombay, India, 1998, pp. 839–846.

[4] N. Sochen, R. Kimmel, and R. Malladi, "A general framework for low level vision," *IEEE Transactions on Image Processing*, pp. 310–318, March 1998.

[5] S. Di Zenzo, "A note on the gradient of a multi-image," *Computer Vision, Graphics and Image Processing*, pp. 402–407, 1986.

[6] J. A. Sethian, "A marching level set method for monotonically advancing fronts," *Proc. Nat. Acad. Sci.*, vol. 93, no. 4, 1996.

[7] Alon Spira, Ron Kimmel, and Nir Sochen, "Efficient beltrami flow using a short time kernel," in *Proceedings of the 4th International Conference on Scale-Space Methods in Computer Vision*, 2003, pp. 511–522.