AN ANISOTROPIC DIFFUSION WITH MEANINGFUL SCALE PARAMETER

Hae Yong Kim

Universidade de São Paulo, Escola Politécnica, Dept. Eng. Sistemas Eletrônicos,
Av. Prof. Luciano Gualberto, tr. 3, 158, 05508-900, São Paulo, SP, Brazil, e-mail: hae@lps.usp.br.

ABSTRACT

The well-known anisotropic diffusion (a.k.a. Perona-Malik equation, nonlinear diffusion, or diffusion partial differential equation – PDE) is widely used in image segmentation, filtering and edge detection. The behavior of the diffusion highly depends on the appropriate choice of the gradient thresholding scale parameter K. However, it seems that no clear relationship between the parameter K and the output image has ever been established, and hence the choice of K is a guesswork. This paper establishes an explicit connection between the parameter K and the number ν of edge-elements (edgels) of the filtered image. Let us define that the frontier between two neighboring pixels (p, q) is an edgel if $|I(p)-I(q)| > \tau$, where $I(p)$ is the image intensity at p and τ is a constant. This connections is valid only for the spatio-temporally discretized diffusion. In our approach, the user specifies the desired ν. From this parameter, an appropriate K is automatically computed in every iteration, so that the final filtered image has ν edgels. Using this approach, the diffusion converges to a nontrivial piecewise constant image, whenever a feasible parameter ν is specified.

1. INTRODUCTION

Linear scale space is a theory introduced by Witkin [1] and used to process an image in multiple resolutions. In this theory, Gaussian low-pass filters process the original fine-scale image, generating simplified coarse-scale images. Unfortunately, coarse-scale images generated by Gaussian filters present blurred edges that do not spatially match the original edges.

In order to keep important edges sharp and spatially fixed, while filtering noise and small edges, Perona and Malik has introduced the nonlinear scale-space [2]. This theory uses the anisotropic diffusion to simplify the original image. Recently, the relations between the anisotropic diffusion and the robust statistics have been established, leading to the robust anisotropic diffusion (RAD) that preserves sharper boundaries than previous techniques [3].

The behavior of the anisotropic diffusion depends on two parameters: the artificial time parameter t and the gradient thresholding parameter K. However, if the discretized diffusion is iterated until the convergence ($t \to \infty$), the output image depends only on K. Thus, the appropriate choice of K is essential to obtain a conveniently filtered image. Though, it seems that no clear relationship between the parameter K and the output image has ever been established, and hence the choice of K is a guesswork.

Some papers have proposed to automatically choose K [3, 4]. We think that such an automated strategy can be applied only to some specific application, because the amount of desired filtering is a user's choice and no automated system can predict the user's mind.

This paper proposes another approach. It establishes an explicit connection between the parameter K and the number ν of edge-elements (edgels) of the filtered image. Let us define that the frontier between two neighboring pixels (p, q) is an edgel if $|I(p) - I(q)| > \tau$, where $I(p)$ is the image intensity at p and τ is a positive constant. For quantized piecewise constant images, τ is usually zero. This connections is valid only for the spatio-temporally discretized anisotropic diffusion.

In our approach, the user specifies the desired ν. This quantity is roughly equivalent to the sum of all edge lengths in the filtered image (distance unit in pixels). From the specified ν, an appropriate parameter K is automatically computed in every diffusion iteration, so that the final filtered image has ν edgels. Note that in each iteration, a different K must be estimated and used. Using this approach, the diffusion converges to a nontrivial piecewise constant image, whenever a feasible parameter ν is specified.

2. ANISOTROPIC DIFFUSION

Perona and Malik defined their anisotropic diffusion as [2]:

$$\frac{\partial I(x,y,t)}{\partial t} = \operatorname{div}\left[g\left(\|\nabla I(x,y,t)\|\right)\nabla I(x,y,t)\right] \qquad (1)$$

using the original image $I(x, y, 0) : \mathbb{R}^2 \to \mathbb{R}^+$ as the initial condition, where t is an artificial time parameter and g is

This work was supported in part by FAPESP under grant 2003/13752-9 and by CNPq under grant 305065/2003-3.

an "edge-stopping" function. They suggested using one of the two edge-stopping functions below (all edge-stopping functions $g_i(x)$ presented in this paper have been dilated and scaled so that $g_i(0) = 1$ and their "influence functions" $\psi_i(x) = x g_i(x)$ have local maxima at $x = K$):

$$g_1(x) = \left[1 + \frac{x^2}{K^2}\right]^{-1}, \quad g_2(x) = \exp\left[-\frac{x^2}{2K^2}\right]. \quad (2)$$

The right choice of the edge-stopping function g can greatly affect the extent to which discontinuities are preserved. Black et al. [3] used the robust estimation theory to choose a better edge-stopping function, called Tukey's biweight:

$$g_3(x) = \begin{cases} \left[1 - \frac{x^2}{5K^2}\right]^2, & \frac{x^2}{5} \leq K^2 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The diffusion that uses this edge-stopping function is called robust anisotropic diffusion (RAD) and this is the edge-stopping function adopted in this paper.

Perona and Malik [2] discretized spatio-temporally their anisotropic diffusion equation (1) as:

$$I(s,t+1) = I(s,t) + \frac{\lambda}{|\eta_s|} \sum_{p \in \eta_s} g(|\nabla I_{s,p}(t)|) \nabla I_{s,p}(t) \quad (4)$$

where $I(s,t)$ is a discretely sampled image, s denotes the pixel position in a discrete 2-D grid, $t \geq 0$ now denotes discrete time steps, the constant λ determines the rate of diffusion (usually, $\lambda = 1$), and η_s represents the set of spatial neighbors of pixel s. For 1-D signals, usually two neighbors are considered: *left* and *right*, except at signal boundaries where only one neighbor must be considered. For 2-D images, usually 4-neighborhood is used: *north*, *south*, *west* and *east*, except at the image boundaries. Perona and Malik approximated the image gradient magnitude in a particular direction at iteration t as:

$$\nabla I_{s,p}(t) = I(p,t) - I(s,t), p \in \eta_s. \quad (5)$$

Black et al. [3] have noticed that the value K_m where the influence function $\psi_i(K_m) = K_m g_i(K_m)$ is maximal determines the threshold between homogeneous regions and edges. If $\|\nabla I_{s,p}(t)\| > K_m$, the frontier between pixels s and p is considered by the diffusion as an edgel to be preserved. And if $\|\nabla I(x,y)\| < K_m$, pixels s and p are considered to belong to the same homogeneous region and the difference between them (possibly due to noises) is gradually suppressed by the diffusion.

3. GRADIENT HISTOGRAM-BASED DIFFUSION

3.1. Automated Selection of K

We list below some of criteria described in the literature to choose an appropriate parameter K:

1. Set K by hand at some fixed value [2].

2. Use the "noise estimator" described by Canny [5]. The histogram of the absolute values of the gradient throughout the image is computed. Then K is set equal to the 90% value of its integral at every iteration [2].

3. Use tools from robust statistics to automatically estimate the "robust scale" σ_e of an image I: $K = \sigma_e = 1.4826 \ \mathrm{median}_I[| \ \|\nabla I\| - \mathrm{median}_I(\|\nabla I\|) \ |]$. It is not clear if K is fixed or should be estimated at every iteration [3].

4. Estimate the noise at every iteration using the difference between the average intensities of images filtered by morphological opening and closing: $K = \mathrm{average}(I \circ e) - \mathrm{average}(I \bullet e)$, where e is a structuring element [4].

5. Use the p-norm of the image to estimate the noise: $K = (\sigma \|I\|_p)/m$, where σ is a constant proportional to the image average intensity and m is the number of image pixels [4].

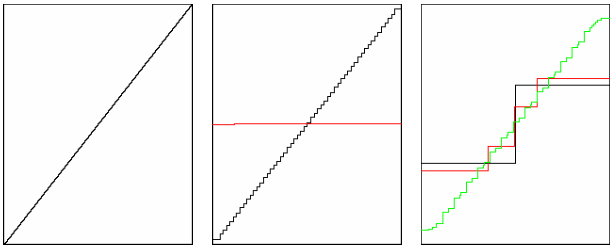
All above criteria try to find a parameter K to appropriately separate edges from noises. However, we argue that the classification of a pair of neighboring pixels in "edge" or "noise" depends not only on the image I to be filtered, but also on the desired scale of the filtered image. If the user wants to obtain a coarse-scale image with only the key edges, a large K should be specified. And if the user wants to obtain a fine-scale image with all the detailed edges, a small K should be chosen. Evidently, no automated process can guess the user's mind.

3.2. An Intuitive Example

We propose a different approach, named Gradient Histogram-based Anisotropic Diffusion (GHAD). GHAD establishes the connection between the parameter K and the number ν of edgels of the filtered image. GHAD is based on the criterion 2. Although this criterion was proposed in the original Perona and Malik's paper, it seems that many of its consequences remain unexplored.

To explain intuitively our approach, let us consider a simple example of 1-D signal processing. Figure 1(a) shows a signal with 201 samples (and consequently 200 possible edgels) that slowly grows from 0 to 255.

We always normalize the original grayscale signal/image from integer intensity range $[0, ..., 255]$ to floating-point range $[0, 1]$ before performing the diffusion. The diffusion is processed using floating-point operations and variables, in order to minimize rounding errors. The obtained filtered floating-point signal/image is converted back to the integer range and saved.



(a) Original signal. (b) Filtered by RAD. (c) Filtered by GHAD.

Fig. 1. 1-D signal processed by RAD and GHAD (see text for explanation).

Filtering the signal 1(a) by RAD with fixed K until the convergence, the filtered signal is either almost equal to the original (black signal of figure 1(b), $K = 0.00585$) or a constant signal (red, $K = 0.00586$). Whenever $K \leq 0.00585$, the first kind of signal is obtained, and whenever $K \geq 0.00586$ the second kind of signal is generated. The conventional anisotropic diffusion with fixed K cannot deal appropriately with smooth edges.

Using GHAD, the user can easily specify the desired number of edgels of the output signal. Figure 1(c) shows the original signal processed by GHAD, using $\nu = 1$ (black), $\nu = 3$ (red) and $\nu = 60$ (green).

3.3. The Proposed Technique

In GHAD, the user specifies the desired number ν of edgels in the final filtered signal/image. Alternatively, the user can specify the proportion $\varpi = \nu/n$, where n is the number of possible edgels of the image.

In each diffusion iteration, GHAD computes the absolute gradient values throughout the image. In the example above, $n = 200$ absolute differences between neighboring pixels are computed. These values are sorted in increasing order, generating the ordered sequence $\vec{v} = (v_0, ..., v_{n-1})$.

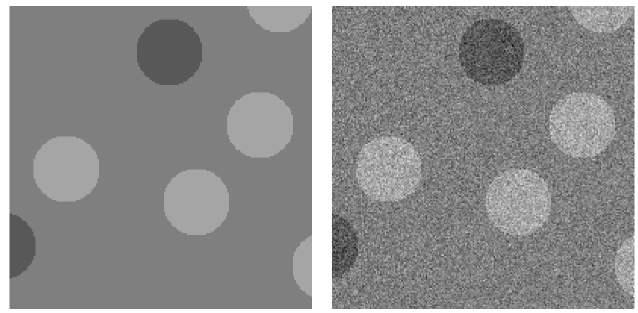
In order to make the output image to have ν edgels, we set $K = v_{n-1-\nu}$ in every iteration. With this setting, the diffusion considers ν frontiers between neighboring pixels as edgels to be preserved, and $n - \nu$ frontiers as homogeneous regions where the noise must be suppressed.

The sequence $\vec{v}$ does not need to be completely sorted. It is enough to know the value of the element $v_{n-1-\nu}$ of the increasing sequence. This value can be efficiently computed in average time $O(n)$ using the algorithm that computes the k-th smallest element of a sequence, derived from the quicksort. The description of this algorithm can be found in many introductory books on algorithms, for example, in [6].

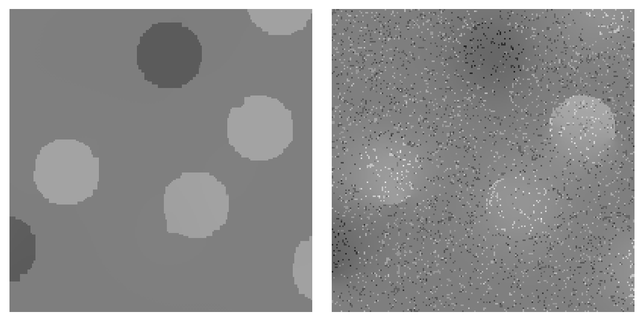
Usually, the anisotropic diffusion uses 4-neighborhood for 2-D images and 6-neighborhood for 3-D volumes. Some care must be taken in 2-D/3-D GHAD implementation, to assure that each frontier between neighboring pixels is computed once and only once in $\vec{v}$. For example, a 3×3 image has 12 possible edgels, and each one of 12 gradient magnitudes must appear once and only once in $\vec{v}$.

4. EXPERIMENTAL RESULTS

4.1. Restoration of Piecewise Constant Images



(a) Original image. (b) Corrupted by Gaussian noise.



(c) Restored by GHAD, PSNR = 40dB. (d) Restored by RAD, PSNR = 25dB.

Fig. 2. Restoration of a piecewise constant image corrupted by Gaussian noise.

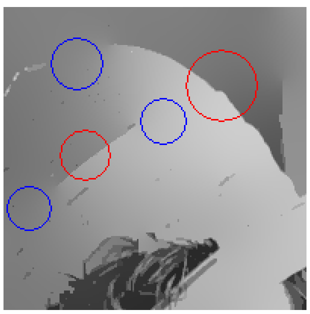
Figure 2 depicts the restoration of a piecewise constant image corrupted by Gaussian noise, using GHAD and RAD. The original image 2(a) has background graylevel 128, with circles with intensities 89 and 166. This image was converted to floating-point range $[0, 1]$, and the zero-mean Gaussian noise with standard deviation 0.08 was added, resulting in figure 2(b).

The original image 2(a) has $\varpi = 0.54\%$ of edgels and consequently this is the best restoration parameter. Filtering the noisy image 2(b) by GHAD with $\varpi = 0.54\%$ (number of iterations $t = 1000$), the image 2(c) was obtained (PSNR = 40.11 dB). GHAD did not succeed to perfectly restore diagonal edges, probably due to the 4-neighborhood spatial discretization.

Filtering the noisy image 2(b) by RAD with K fixed at 0.04 ($t = 1000$), the image 2(d) was obtained (PSNR = 24.69 dB). RAD could neither eliminate noises nor preserve edges sharp. RAD with another K will not do a better restoration because if a larger K is chosen, the edges will become more blurred; and if a smaller K is chosen, more noise will remain unfiltered.

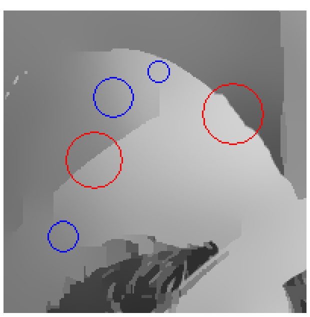
4.2. Filtering Natural Images with GHAD

(a) Original "Lenna" image.



(b) Filtered by RAD, $t = 1000$.

(c) Filtered by RAD, $t \to \infty$.



(d) Filtered by GHAD, $t = 500$.

(e) Filtered by GHAD, $t \to \infty$.

Fig. 3. Image "Lenna" processed by RAD ($K = 0.022$) and GHAD ($\varpi = 5\%$).

Figure 3 depicts "Lenna" image 3(a) filtered by RAD and GHAD. RAD's parameters $K = 0.022$ and GHAD's parameter $\varpi = 5\%$ were chosen so that the final filtered images 3(c) and 3(e) have roughly the same number of edgels.

RAD eliminates many important edges (red circles in figure 3(b)) when iterated until the convergence (figure 3(c), $t = 30000$). This happens because the grayscale intensity leaks continously through smooth edges (blue circles in figure 3(b)). As we noted in section 3, the anisotropic diffusion with fixed K cannot deal appropriately with slowly varying edges. A smooth edge will be either eliminated or preserved unaltered (see figure 1).

GHAD, on the contrary, preserves important edges sharp (red circles in figure 3(d)) even when iterated until the convergence (figure 3(e), $t = 40000$). GHAD converts, if necessary, smooth edges (blue circles in figure 3(d)) in sharp edges in order to clearly delimit constant regions. Unfortunately, GHAD shows strong preference for new horizontal and vertical edges over new diagonal edges. This is probably due to the 4-neighborhood spatial discretization.

5. CONCLUSIONS AND FUTURE WORKS

This paper has introduced a new variety of anisotropic diffusion named GHAD (Gradient Histogram-based Anisotropic Diffusion). GHAD produces a piecewise constant image with ν edge-elements when iterated until the convergence. We have shown that GHAD outperforms the conventional robust anisotropic diffusion (RAD) in the restoration of a piecewise constant image corrupted by Gaussian noise. We have also shown that the conventional RAD cannot appropriately filter smooth edges, while GHAD converts these edges into sharp ones. In this conversion, GHAD shows preference for new horizontal/vertical edges over diagonal ones. We are researching spatial discretizations that do not have preferences for any particular direction.

6. REFERENCES

[1] A. P. Witkin, "Scale-Space Filtering," *Proc. 8th Int. Joint Conf. Art. Intelligence*, vol. 2, pp. 1019–1022, 1983.

[2] P. Perona and J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," *IEEE. Trans. Patt. Anal. and Machine Intell.*, vol. 12, no. 7, pp 629–639, 1990.

[3] M. J. Black, G. Sapiro, D. H. Marimont and D. Hegger, "Robust Anisotropic Diffusion," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 421–432, Mar. 1998.

[4] F. Voci, S. Eiho, N. Sugimoto and H Sekiguchi, "Estimating the Gradient Threshold in the Perona-Malik Equation," *IEEE Signal Processing Magazine*, pp. 39–46, May 2004.

[5] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679–698, 1986.

[6] R. Sedgewick, *Algorithms in C++*, Addison-Wesley, 1992.