ANISOTROPIC VECTOR DIFFUSION IN IMAGE SMOOTHING

Zeyun Yu and *Chandrajit Bajaj*

Department of Computer Sciences, University of Texas at Austin, Austin, Texas 78712, USA
{zeyun, bajaj}@cs.utexas.edu

ABSTRACT

Anisotropic diffusion has been widely used in image processing for its efficiency of smoothing the noisy images while preserving the sharp edges. In this paper we will explore a general version of anisotropic diffusion schemes for vector-valued images, based on the polar-coordinate representation of the vectors. As an example, we will apply our method to color images and show its ability of edge-preserving smoothing on vector-valued images.

1. INTRODUCTION

Image smoothing, as a preprocessing step, plays a very important role in image processing and other image-related research areas. Conventional Gaussian filter [11] can be viewed as an isotropic diffusion, making the entire image uniformly smoothed and thus blurring the image edges. To remedy this problem, Perona and Malik [1] proposed a nonlinear anisotropic version of the heat diffusion equations, in which the conduction coefficient is a decreasing function of the magnitude of the image gradient:

$$\frac{\partial I}{\partial t} = \nabla \cdot (c(|\nabla I|)\nabla I) \quad (1)$$

The elegant property of the anisotropic diffusion scheme is that it preserves all important edge features while smoothing out image noises. Due to this reason, this approach has drawn a lot of attention since it was proposed (see [3, 4, 5] for examples). While lots of research work has been devoted to the theoretical properties and practical applications of this technique, its extension to vector-valued images has also been discussed by many authors [6, 7, 8, 9]. The most straightforward way to extend the anisotropic diffusion to vector-valued images is to simply apply equation (1) separately to each component of the vectors over the image domain. However, it was shown that this simple scheme did not work very well when different conduction coefficients were used for each component of the vectors [10]. Hence, most anisotropic diffusion schemes for vector-valued images are using a common diffusion tensor which combines the information from all components of the vectors.

In the present paper we will consider a new type of anisotropic vector diffusion, in which we build the diffusion equations using the polar-coordinate representation (magnitude and orientation separately) of the vectors instead of the conventional Cartesian-coordinate representation as seen in most anisotropic diffusion schemes. As we will see in next section, the diffusion scheme based on conventional Cartesian-coordinate representation is actually a special case of our diffusion method based on polar-coordinate representation of the vectors. Furthermore, for some particular applications, it is more convenient to consider the diffusions separately on magnitude and orientation of the vectors. For example, we have successfully applied this technique to gradient vector diffusion for image segmentation, where only 2D vector diffusion were discussed [2]. Other applications include extraction of median axis of gray-scale images and construction of Voronoi diagrams based on gradient vector diffusion (see [13] for details of gradient vector diffusion). However, those topics are out of the scope of this paper and in the following we will concentrate on the algorithm of 3D anisotropic vector diffusion as an extension of our past work [2]. But as an example, we will consider the application of our method in the color image smoothing where the RGB values are viewed as a 3D vector at each image pixel.

We organize this paper as follows. In next section we will describe the details of the anisotropic vector diffusion technique (in 3D case), based the polar-coordinate representation of vectors. And then, in section 3, we will briefly discuss the application of our algorithm in color image smoothing and some results will also be shown there. Finally we give our conclusion and future work.

2. ANISOTROPIC VECTOR DIFFUSION

2.1. Analysis

As we know, equation (1) says that the new value $I^{new}(x, y)$ at point (x, y) is determined by the old value $I^{old}(x, y)$ and the weighted average of the image gradients in the directions from (x, y) to each of its neighbors. Hence, to study the new vector values at a point, we just need to study how each of the neighboring vectors affects the vector being con-

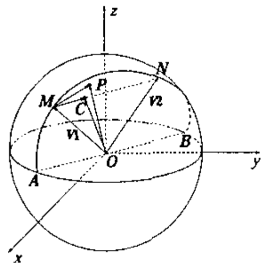
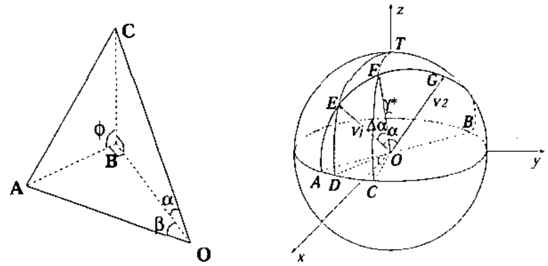


Fig. 1. Illustration of different diffusion schemes



(a) for Lemma *(b) for Theorem*

Fig. 2. Illustration of Lemma and Theorem

sidered. Without loss of generality, we will consider a vector $\vec{v}_1$ at point (x, y) and one of its neighboring vectors $\vec{v}_2$ (as seen in *fig.*1). The diffusion scheme, based on Cartesian-coordinate representation of the vectors, makes $\vec{v}_1$ move along the direction from $\vec{v}_1$ to $\vec{v}_2$ and the new vector $\vec{OC}$ is a linear combination of $\vec{v}_1$ and $\vec{v}_2$ if we are using a common conduction coefficient for the diffusion equations on each component of the vectors. On the other hand, the diffusion scheme based on polar-coordinate representation makes $\vec{v}_1$ move towards $\vec{v}_2$ within the plane determined by $\vec{v}_1$ and $\vec{v}_2$ but the resulting vector is a nonlinear combination of $\vec{v}_1$ and $\vec{v}_2$ (shown as $\vec{OP}$ in *fig.*1). As we will see in the following, the diffusion scheme based on conventional Cartesian-coordinate representation can actually be viewed as a special case of our diffusion scheme if we carefully choose the coefficients for diffusion equations on magnitude and orientation of the vectors, such that the resulting vector $\vec{OP}$ is exactly the same as the vector $\vec{OC}$.

In the following we will consider the diffusion schemes separately for magnitude r and orientation θ and γ, where the transformation from polar coordinate (r,θ,γ) to Cartesian coordinate (x, y, z) is given by:

$$\begin{cases} x = r \cdot cos(\gamma) \cdot cos(\theta) \\ y = r \cdot cos(\gamma) \cdot sin(\theta) \\ z = r \cdot sin(\gamma) \end{cases}$$

for $r \geq 0$, $\gamma \in [-\pi/2, \pi/2]$ and $\theta \in (-\pi, \pi]$.

2.2. Diffusion Equation on r

The diffusion equation on magnitude r is the same as classic anisotropic diffusion of scalar values:

$$\frac{\partial r}{\partial t} = \nabla(c_1(\vec{v}_1, \vec{v}_2)\nabla r), \qquad (2)$$

where $\vec{v}_1$ is the vector being considered and $\vec{v}_2$ is the neighboring vectors. The coefficient $c_1(\vec{v}_1, \vec{v}_2)$ can be chosen as a decreasing function of $|\nabla r|$ or $|\,\|\vec{v}_1\| - \|\vec{v}_2\|\,|$ as seen in classic anisotropic diffusion scheme. But usually it is better to choose $c_1(.)$ as a decreasing function of $\|\vec{v}_1 - \vec{v}_2\|$.

2.3. Diffusion Equations on θ and γ

Similar to the conventional Cartesian-coordinate based scheme, the diffusion equations on θ and γ cannot be diffused separately to guarantee that the resulting vector should be within the plane determined by the vector $\vec{v}_1$ and one of its neighboring vectors $\vec{v}_2$. Assume the angle between $\vec{v}_1$ and $\vec{v}_2$ is α. Then we can compute how much $\vec{v}_1$ will move to the new position $\vec{v}^*$ (or $\vec{OF}$ as seen in *fig.2(b)*) along the direction from $\vec{v}_1$ to $\vec{v}_2$ by using a similar strategy as in equation (2) but the coefficient $c(.)$ may be different from $c_1(.)$. Suppose the value we got is $\Delta\alpha$. Now the problem we need to solve is that how to represent the new vector $\vec{v}^*$ in polar coordinates using the known $\vec{v}_1$ and $\vec{v}_2$. Keep in mind that here we are considering only one of the neighbors. The final position of vector $\vec{v}_1$ should be determined by the weighted average of all such $\vec{v}^*$'s affected by all the neighboring vectors. In the following we first give a *Lemma* without proof.

Lemma: Consider a point O in 3D space with three lines (OA, OB and OC) passing through it (as shown in *fig.2(a)*). Suppose the planar angle between two planes AOB and BOC is ϕ, and $\angle AOB = \beta$, $\angle BOC = \alpha$. Let γ be $\angle AOC$. Then we have:

$$cos(\gamma) = cos(\alpha)cos(\beta) + sin(\alpha)sin(\beta)cos(\phi). \qquad (3)$$

Now let's go back to the problem of how to compute the polar coordinates (θ^*, γ^*) of the diffused vector $\vec{v^*}$, given a vector $\vec{v}_1 : (\theta_1, \gamma_1)$ and one of its neighbors $\vec{v}_2 : (\theta_2, \gamma_2)$ (as shown in fig.2(b)). First of all, we can easily compute the angle α between $\vec{v}_1$ and $\vec{v}_2$:

$$\alpha = cos^{-1}(cos(\gamma_1)cos(\gamma_2)cos(\theta_1 - \theta_2) + sin(\gamma_1)sin(\gamma_2)). \qquad (4)$$

Consider the three lines OE, OT, OF. Assume the planar angle between plane OEF and plane OET is φ. Then by *Lemma* we have:

$$cos(\tfrac{\pi}{2} - \gamma^*) = cos(\tfrac{\pi}{2} - \gamma_1)cos(\Delta\alpha) + sin(\frac{\pi}{2} - \gamma_1)sin(\Delta\alpha)cos(\varphi), \qquad (5)$$

or equivalently,

$$sin(\gamma^*) = sin(\gamma_1)cos(\triangle\alpha) + cos(\gamma_1)sin(\triangle\alpha)cos(\varphi). \quad (6)$$

Similarly consider the three lines OE, OT, OG and we have:

$$cos(\tfrac{\pi}{2} - \gamma_2) = cos(\tfrac{\pi}{2} - \gamma_1)cos(\alpha) + sin(\frac{\pi}{2} - \gamma_1)sin(\alpha)cos(\varphi), \quad (7)$$

or equivalently,

$$sin(\gamma_2) = sin(\gamma_1)cos(\alpha) + cos(\gamma_1)sin(\alpha)cos(\varphi). \quad (8)$$

By calculating $sin(\alpha) \times (6) - sin(\triangle\alpha) \times (8)$ we have:

$$sin(\gamma^*)sin(\alpha) - sin(\gamma_2)sin(\triangle\alpha) = sin(\gamma_1)sin(\alpha - \triangle\alpha). \quad (9)$$

And thus,

$$sin(\gamma^*) = \frac{sin(\gamma_2)sin(\triangle\alpha) + sin(\gamma_1)sin(\alpha - \triangle\alpha)}{sin(\alpha)}. \quad (10)$$

Notice that equation (10) is true only if $\alpha \neq 0$ and $\alpha \neq \pi$, or in other words, vectors $\vec{v}_1$ and $\vec{v}_2$ should be independent on each other.

Now let's consider the three lines OE, OF, OT again. Obviously the planar angle between plane OET and plane OFT is $\triangle\theta$, where $\triangle\theta = |\theta_1 - \theta^*|$ (Remember that $\angle COD$ is sometimes not equal to $|\theta_1 - \theta^*|$ because θ is periodically defined on $(-\pi, \pi]$. But for simplicity we will ignore this in the following). By *Lemma* we have:

$$cos(\triangle\alpha) = cos(\tfrac{\pi}{2} - \gamma_1)cos(\tfrac{\pi}{2} - \gamma^*) + sin(\frac{\pi}{2} - \gamma_1)sin(\frac{\pi}{2} - \gamma^*)cos(\triangle\theta), \quad (11)$$

or equivalently,

$$cos(\triangle\alpha) = sin(\gamma_1)sin(\gamma^*) + cos(\gamma_1)cos(\gamma^*)cos(\triangle\theta). \quad (12)$$

So we get:

$$cos(\triangle\theta) = \frac{cos(\triangle\alpha) - sin(\gamma_1)sin(\gamma^*)}{cos(\gamma_1)cos(\gamma^*)}. \quad (13)$$

Notice that equation (13) is true only if $\gamma_1 \neq \pm\frac{\pi}{2}$ and $\gamma^* \neq \pm\frac{\pi}{2}$ which means $\vec{v}_1$ and $\vec{v^*}$ can not be parallel to axis z. After we know $\triangle\theta$, we can easily find out θ^* by either $\theta_1 + \triangle\theta$ or $\theta_1 - \triangle\theta$ (depending on which side vector $\vec{v}_2$ lies on, with respect to vector $\vec{v}_1$).

From (10) and (13) we have the following theorem for determining the new position of a vector "attracted" by one of its neighbors:

Theorem: Given a vector $\vec{v}_1 : (\theta_1, \gamma_1)$ and one of its neighbors $\vec{v}_2 : (\theta_2, \gamma_2)$, the polar coordinate of the diffused vector $\vec{v}^* : (\theta^*, \gamma^*)$ can be described by:

$$\begin{cases} \gamma^* = sin^{-1}\left(\frac{sin(\gamma_2)sin(\triangle\alpha) + sin(\gamma_1)sin(\alpha - \triangle\alpha)}{sin(\alpha)}\right) \\ \theta^* = \theta_1 \pm cos^{-1}\left(\frac{cos(\triangle\alpha) - sin(\gamma_1)sin(\gamma^*)}{cos(\gamma_1)cos(\gamma^*)}\right) \end{cases} \quad (14)$$

where α is the angle between $\vec{v}_1$ and $\vec{v}_2$, and $\triangle\alpha$ is the angle difference by which $\vec{v}_1$ moves towards $\vec{v}_2$. ∎

As a summary of this section, let's suppose we are going to determine the new position of the vector $\vec{v} : (\theta, \gamma)$ at (x, y). First we need to find out all the neighbors of (x, y) using 4-neighbor scheme (or other schemes). Each of the neighboring vectors (denoted by $\vec{v}_i$, $i = 1, 2, 3, 4$) makes the vector $\vec{v}$ rotate along a specific direction and thus we get: $\triangle\gamma_i, \triangle\theta_i, i = 1, 2, 3, 4$. $\triangle\gamma_i, \triangle\theta_i$ are calculated using $\triangle\gamma_i = \gamma_i^* - \gamma$ and $\triangle\theta_i = \theta_i^* - \theta$, where γ_i^* and θ_i^* are obtained by equation (14). Then the new position of vector $\vec{v}$ after diffusion (for each iteration) can be determined by:

$$\begin{cases} \gamma^{new} = \gamma^{old} + \lambda \sum_{i=1}^{4} \triangle\gamma_i \\ \theta^{new} = \theta^{old} + \lambda \sum_{i=1}^{4} \triangle\theta_i \end{cases} \quad (15)$$

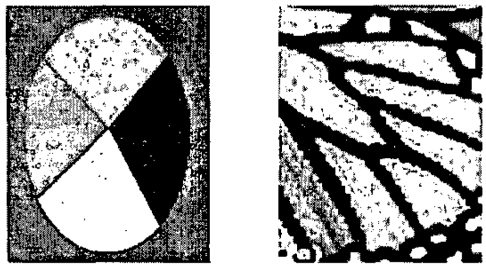
where $0 \leq \lambda \leq \frac{1}{4}$ for the solutions to be stable [1]. And r^{new} is separately determined by equation (2).

3. EXAMPLE: COLOR IMAGE SMOOTHING

In previous section we described a general version of anisotropic vector diffusion based on the polar-coordinate representation of a vector. This algorithm can be applied to various types of vector-valued images for smoothing. In this section we will consider the color images and show some smoothing results using our method.

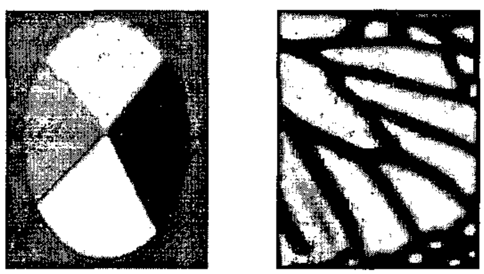
We will take the simple RGB color model in our implementation. That means the RGB values at each pixel will be viewed as a vector for that pixel. Some other authors considered other more complicated color models. For example, Lucchese and Mitra [12] considered anisotropic diffusion on the complex chromaticity and lightness of the color images. This idea, in some sense, is very similar to our polar-coordinate representation of a vector if we think the magnitude as the lightness and the orientation as the chromaticity. However, their approach was designed in particular for color image smoothing, while our vector diffusion method can be used not only for color image smoothing but also for other purposes such as image segmentation and skeleton extraction.

Fig.3 shows two noisy color images for testing. One is an artificial image and the other is a real image. Both images



(a) (b)

Fig. 3. *Testing images added with randomly generated noises: (a) is an artificial image and (b) is a real image.*



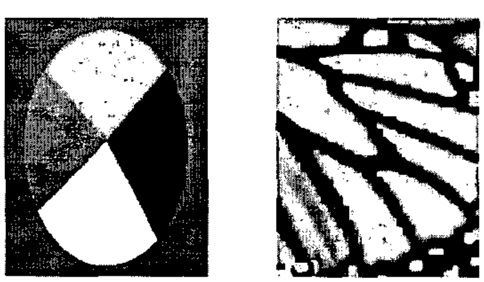
(a) (b)

Fig. 4. *Results by using isotropic diffusion*

are degraded with randomly generated noises. The results of isotropic diffusion on both images are shown in *fig.4(a)* and *(b)*, respectively. It is very obvious that the edges are blurred although the images are smoothed quite well elsewhere. As compared to the isotropic diffusion, our anisotropic vector diffusion scheme smoothes out the noises quite well while sharp edges are preserved (see *fig.5(a)* and *(b)*).

4. CONCLUSION

In this paper we presented a general version of anisotropic diffusion for vector-valued images. Our scheme is based on the polar-coordinate representation of vectors. As an example we tested our algorithm on color image for its ability of smoothing out noises. Experiments showed that our scheme had the same elegant property of conventional anisotropic diffusion of scalar images. We have already successfully applied our method to gradient vector diffusion for the purpose of image segmentation. Our future work will concentrate on the relationship between our anisotropic vector diffusion and construction of Voronoi diagrams as well as extraction of median axis.



(a) (b)

Fig. 5. *Results by using our anisotropic vector diffusion*

5. REFERENCES

[1] P.Perona and J.Malik, "Scale-space and edge detection using anisotropic diffusion", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 7, pp. 629-639, 1990.

[2] Z.Yu and C.Bajaj, "Image segmentation using gradient vector diffusion and region merging", *Proc. 16th International Conference on Pattern Recognition (ICPR'02)*, Quebec, Canada, August 11-15, 2002.

[3] J.Weickert, "Theoretical foundations of anisotropic diffusion in image processing", *Computing*, Vol. 11, pp. 221-236, 1996.

[4] J.Weickert, "Anisotropic diffusion in image processing", *ECMI Series, Teubner, Stuttgart, 1998. ISBN 3-519-02606-6.*

[5] M.J.Black and G.Sapiro, "Robust anisotropic diffusion", *IEEE Trans. on Image Processing*, Vol. 7, No. 3, pp. 421-432, 1998.

[6] R.T.Whitaker, "Vector-valued diffusion", *Geometry Driven Diffusion in Computer Vision* (Edited by B.Romeny), pp. 93-134, 1994.

[7] G.Sapiro and D.Ringach, "Anisotropic diffusion of multivalued images with applications to color filtering", *IEEE Trans. on Image Processing*, Vol. 5, No. 11, pp. 1582-1586, 1996.

[8] G.Sapiro, "Vector-valued active contours", *Proc. Computer Vision and Pattern Recognition*, San Francisco, June 1996.

[9] J.Weickert, "Coherence-enhancing diffusion of color images", *Proc. 7th National Symposium on Pattern Recognition and Image Analysis*, Vol. 1, pp. 239-244, April, 1997.

[10] J.Weickert, "A review of nonlinear diffusion filtering", *Scale-Space Theory in Computer Vision* (Edited by B.Romeny *et.al.*), Vol. 1252, Springer, Berlin, pp. 3-28, 1997.

[11] R. Haralick and L. Shapiro, *Computer and Robot Vision*, Addison-Wesley Publishing Company, 1992, Vol. 1, Chap. 7.

[12] L.Lucchese and S.K.Mitra, "Color image segmentation through independent anisotropic diffusion of complex chromaticity and lightness", *Proc. IEEE International Conference on Image Processing (ICIP'01)*, Greece, Sept.2001.

[13] C. Xu and J.L.Prince, "Snakes, shapes, and gradient vector flow", *IEEE Trans. Image Processing*, Vol. 7, No. 3, pp. 359-369, 1998.