Fast Difference Schemes for Edge Enhancing Beltrami Flow*

R. Malladi and I. Ravve

Mail Stop: 50A-1148, 1 Cyclotron Road
Lawrence Berkeley National Laboratory
Computing Science Department
University of California, Berkeley, CA 94720
{malladi,ravve}@math.lbl.gov

Abstract. The Beltrami flow [13,14] is one of the most effective denoising algorithms in image processing. For gray-level images, we show that the Beltrami flow equation can be arranged in a reaction-diffusion form. This reveals the edge-enhancing properties of the equation and suggests the application of additive operator split (AOS) methods [4,5] for faster convergence. As we show with numerical simulations, the AOS method results in an unconditionally stable semi-implicit linearized difference scheme in $2D$ and $3D$. The values of the edge indicator function are used from the previous step in scale, while the pixel values of the next step are used to approximate the flow. The optimum ratio between the reaction and diffusion counterparts of the governing PDE is studied, in order to achieve a better quality of segmentation. The computational time decreases by a factor of ten, as compared to the explicit scheme. For 2D color images, the Beltrami flow equations are coupled, and do not yield readily to the AOS technique. However, in the proximity of an edge, the cross-products of color gradients nearly vanish, and the coupling becomes weak. The principal directions of the edge indicator matrix are normal and tangent to the edge. Replacing the action of the matrix on the gradient vector by an action of its eigenvalue, we reduce the color problem to the gray level case with a reasonable accuracy. The scalar edge indicator function for the color case becomes essentially the same as that for the gray level image, and the fast implicit technique is implemented.

Keywords: Beltrami Flow, Unconditionally Stable Schemes, Color Images, Segmentation.

1 Introduction

The main objective in early computer vision is to smooth images without destroying the semantic content, i.e. edges, features, corners, etc. In other words,

* This work was supported by the Director, Office of Science, Office of Advanced Scientific Research, Mathematical, Information, and Computational Sciences Division, U.S. Department of Energy under Contract No. DE-AC03-76SF00098, and LBNL Directed Research and Development Program

A. Heyden et al. (Eds.): ECCV 2002, LNCS 2350, pp. 343–357, 2002.

the boundaries between objects in the image should survive as long as possible along the scale, while homogeneous regions should be simplified and flattened in a rapid way. This is particularly important since the denoising an image is usually a precursor to segmentation and representation which often rely on edge fidelity.

The use of diffusion equations for image processing in computer vision originated with the work of [1] where the authors pre-select a diffusion coefficient function in the image that preserves the edge information. A more rigorous view was achieved with the realization that the iso-intensity contours of an image can be moved under their curvature following the work of Osher and Sethian [6]. This lead to a series of papers starting with the one by Alvarez, Lions and Morel [2], of viewing the images as a set of level contours and moving then under their curvature. Image smoothing by way of level set curvature motion [3,7] thwarts the diffusion in the edge direction, thereby preserving the edge information. The work by Malladi and Sethian [7,8] showed that in addition to this basic approach, a natural stopping criterion can also be chosen to prevent over smoothing a given image. For a comprehensive look at various approaches that rely on geometric diffusion, the reader is referred to [10]. Another crucial idea in [8] was also to diffuse an image by viewing it as a graph of a function and moving it under the mean curvature.

An important question still remained, what is the natural way to treat vector-valued images and images in higher dimensions? An answer to this and other questions was attempted in [13,14] by Sochen, Kimmel, and Malladi. The result is a general mathematical framework for feature-preserving image smoothing that applies seamlessly to gray level, vector-value (color) images, volumetric images, and movies. The main idea is to view images as embedded maps between two Riemannian manifolds and to define an action potential that provides a measure on the space of these maps. The authors in [14] showed that many classical geometric flows emerge as special cases in this view as well as a new flow, the so called Beltrami flow that moves a gray level image under a scaled mean curvature, and also succeeds in finding a natural coupling between otherwise decoupled component-wise diffusion that was often used in the past in vector-valued image diffusion. In the case of gray value images, and by following a different approach, Yezzi in [21] arrived at a similar equation.

Smoothing of noisy images and edge enhancement usually presents a numerical integration of a parabolic PDE with one dimension in scale and two dimensions in space. This is often the most time consuming component of nonlinear image processing algorithms. Smoothing technique governed by the Beltrami flow equation is one of the most effective since it incorporates the edge indicator function, that minimizes diffusion at and across the edges and extensive diffusion elsewhere. On the other hand, solving the equations using explicit methods can be very time consuming due to the scaling and small time step requirement. We aim to build faster methods to solve the Beltrami flow equation in $2D$ gray level and color, and volumetric imagery. we use the method of Additive Operator Split (AOS). This technique was introduced by Weickert [4] for the nonlinear diffusion

flow and later applied by Goldenberg et al. [5] to implement a fast version of the geodesic contour model.

On a different note, the Beltrami flow equation results from minimizing a (natural) generalization of $L2$ Euclidean norm to non-Euclidean manifolds, see [13] for details. This suggests that the governing equation is an "edge-preserving" in contrast to being an "edge-enhancing" flow. In other words, on grey level images, the equation simulates a mean curvature flow scaled by an edge-indicator function, thereby preserving the edge features in scale. In a recent note on the study of intermediate asymptotics of certain commonly used anisotropic diffusion equations, Barenblatt [9] reports that Beltrami flow equation forms a sharp step in the viscinity of edges. In this work, an asymptotic self-similar solution was obtained for a particular case of the Beltrami equation, suggesting an "edge-sharpening" behavior. In the present context, while performing operator splitting, we confirm that Beltrami flow has both edge-preserving and edge-sharpening components.

Our goal is to build a fast and reliable method to solve the Beltrami flow equations and it is based on AOS technique. This paper is organized as follows. In Section 2, we rearrange the governing equation for the Beltrami flow. This approach leads to a semi-implicit linearized difference scheme. In Section 3, we present numerical simulations for $2D$ and $3D$ gray level images. We run the flow to different scales and different relative magnitudes of reaction term vs. diffusion components are considered. In Section 4 we consider the color images where the edge indicator function is expressed by a matrix [11,14]. Equations for two-dimensional images become coupled, and this does not allow us to apply the AOS splitting immediately. However, in the proximity of the edge, the cross-products of color gradients all vanish or almost vanish. The Beltrami operator then becomes weakly coupled, and the principal directions of the edge indicator matrix are normal and tangent to the edge. The action of this matrix on the gradient vector of a specific color component may be replaced by the action of the eigenvalue on this vector. Thus, the problem is reduced to a scalar gray level case with a reasonable degree of accuracy, and the fast implicit smoothing scheme is implemented. Section 5 presents the results of numerical simulation for color images. We summarize this work in Section 6.

2 Implicit Scheme for Gray Level Images

Let us denote by (Σ, g) the image manifold and its metric and by (M, h) the space-feature manifold and its metric, then the map $X : \Sigma \to M$ has the following measure, [16]:

$$S[X^i, g_{\mu\nu}, h_{ij}] = \int d^m\sigma \sqrt{g} g^{\mu\nu} \partial_\mu X^i \partial_\nu X^j h_{ij}(X), \tag{1}$$

where m is the dimension of Σ, g is the determinant of the image metric, $g^{\mu\nu}$ is the inverse of the image metric, the range of indices is $\mu, \nu = 1, \ldots, \dim \Sigma$, and $i, j = 1, \ldots, \dim M$, and h_{ij} is the metric of the embedding space. This is

a natural generalization of the $L2$ norm to manifolds. As an example, a grey level image can be treated as a $2D$ manifold embedded in R^3, i.e. a mapping $X : (x, y) \rightarrow (X^1 = x, X^2 = y, X^3 = U(x, y))$.

Many scale-space methods, linear and non-linear can be shown to be a gradient descent flows of this functional with appropriately chosen metric of the image manifold. The gradient descent equation is $X^i_t = -\frac{1}{\sqrt{g}} \frac{\delta S}{\delta X^i}$. As shown in [13], minimizing the area action in Eqn. 1, with respect to the feature coordinate U, we obtain the following Beltrami flow equation,

$$\dot{U} = \frac{U_{xx}(U_y^2 + 1) - 2U_x U_y U_{xy} + U_{yy}(U_x^2 + 1)}{(1 + U_x^2 + U_y^2)^2}. \tag{2}$$

The nonlinear diffusion equation is the following reaction-diffusion partial differential equation [4,5]

$$\dot{U} \;=\; \nabla \cdot \left(\frac{\nabla U}{g} \right) \;=\; \frac{\partial}{\partial x} \left(\frac{U_x}{g} \right) + \frac{\partial}{\partial y} \left(\frac{U_y}{g} \right) \tag{3}$$

where $g = (1 + U_x^2 + U_y^2)$ is the gradient magnitude.

The Beltrami equation may be reduced to a similar reaction-diffusion form, namely

$$\dot{U} = \nabla \cdot \left(\frac{\nabla U}{2g} \right) + \frac{\nabla^2 U}{2g} = h \nabla^2 U + 1/2 \nabla h \cdot \nabla U \tag{4}$$

where $h = 1/g$, is the edge indicator function. In this form, the Beltrami flow equation is not a "pure" diffusion equation. It has both an (parabolic) edge-preserving and an (hyperbolic) edge-sharpening terms. In addition, the reaction-diffusion form of Eq. (4) hides the mixed derivative U_{xy}, thereby making it conducive to the AOS approach. In other words, the equation can be rearranged into the form $\dot{U} = (A_x + A_y)U$, where A_x and A_y are the following differential operators:

$$A_x \;=\; \frac{\partial}{\partial x} \left(\frac{h}{2} \frac{\partial}{\partial x} \right) + \frac{h}{2} \frac{\partial^2}{\partial x^2} \qquad A_y \;=\; \frac{\partial}{\partial y} \left(\frac{h}{2} \frac{\partial}{\partial y} \right) + \frac{h}{2} \frac{\partial^2}{\partial y^2}. \tag{5}$$

Applying the backward difference formula to the above form we get,

$$\frac{\boldsymbol{U}^{n+1} \;-\; \boldsymbol{U}^n}{\Delta t} \;=\; (\boldsymbol{A}_x \;+\; \boldsymbol{A}_y)\; \boldsymbol{U}^{n+1} \tag{6}$$

The superscript n is related to the present and $n+1$ to the next time step. The subscripts i, j index the discrete pixel location; $\boldsymbol{U}^n_{i,j}$ are known values, and $\boldsymbol{U}^{n+1}_{i,j}$ are to be found. Using $\boldsymbol{U}^{n+1}$ on the right side of Eq. (6) makes the integration scheme implicit and unconditionally stable, namely

$$[\boldsymbol{I} - \Delta t\, (\boldsymbol{A}_x + \boldsymbol{A}_y)\;]\; \boldsymbol{U}^{n+1} \;=\; \boldsymbol{U}^n \tag{7}$$

where $\boldsymbol{I}$ is the identity matrix. Before proceeding in time, we calculate the values of the edge indicator function g, using the known values of $\boldsymbol{U}^n$. Thus, the scheme

is only semi-implicit. Although g depends on the gradient of U, we treat it like a given function of (x, y), making the governing PDE "quasi-linear".

Note that Eq. (7) includes a large bandwidth matrix, because all equations, related to new pixel values $\boldsymbol{U}^{n+1}$ are coupled. Our aim is to decouple the set (7) so that each row and each column of pixels can be handled separately. For this, we re-arrange the equations into the following form:

$$\boldsymbol{U}^{n+1} \;=\; [\boldsymbol{I} - \Delta t\,(\boldsymbol{A}_x + \boldsymbol{A}_y)\,]^{-1}\;\boldsymbol{U}^n \tag{8}$$

Of course, we do not intend to invert the matrix to solve the linear set. This is only a symbolic form used for further derivation. For a small value of Δt, the matrix in the brackets on the right side of Eq. (8) is close to the identity $\boldsymbol{I}$. Thus, its inverse can be expanded into the Taylor series in the proximity of $\boldsymbol{I}$: $[\boldsymbol{I} - \Delta t\,(\boldsymbol{A}_x + \boldsymbol{A}_y)\,]^{-1} \;\approx\; \boldsymbol{I} + \Delta t\,(\boldsymbol{A}_x + \boldsymbol{A}_y)$, where the linear term is retained and the high order terms are neglected. Introducing this form into (8), we get,

$$2\boldsymbol{U}^{n+1} \;=\; (\boldsymbol{I} + 2\Delta t\boldsymbol{A}_x)\,\boldsymbol{U}_n \;+\; (\boldsymbol{I} + 2\Delta t\boldsymbol{A}_y)\,\boldsymbol{U}^n \tag{9}$$

Introducing the notations $\boldsymbol{V} = (\boldsymbol{I} + 2\Delta t\boldsymbol{A}_x)\boldsymbol{U}^n$ and $\boldsymbol{W} = (\boldsymbol{I} + 2\Delta t\boldsymbol{A}_y)\boldsymbol{U}^n$ the solution is simply

$$\boldsymbol{U}^{n+1} \;=\; \frac{\boldsymbol{V} + \boldsymbol{W}}{2} \tag{10}$$

In order to get an implicit scheme, we apply the differential matrix operators $\boldsymbol{A}_x$ and $\boldsymbol{A}_y$ to $\boldsymbol{U}^{n+1}$ (and not to $\boldsymbol{U}^n$), namely

$$(\boldsymbol{I} + 2\Delta t\boldsymbol{A}_x)^{-1}\,\boldsymbol{V} \;=\; \boldsymbol{U}^n \qquad\qquad (\boldsymbol{I} + 2\Delta t\boldsymbol{A}_y)^{-1}\,\boldsymbol{W} \;=\; \boldsymbol{U}^n \tag{11}$$

Following the procedure of expanding the matrix inverses into Taylor series and applying the linearization for small Δt, we finally obtain the equation sets for $\boldsymbol{V}$ and $\boldsymbol{W}$ as follows:

$$(\boldsymbol{I} - 2\Delta t\boldsymbol{A}_x)\,\boldsymbol{V} \;=\; \boldsymbol{U}^n \qquad\qquad (\boldsymbol{I} - 2\Delta t\boldsymbol{A}_y)\,\boldsymbol{W} \;=\; \boldsymbol{U}^n \tag{12}$$

These equations can be solved with either the Dirichlet or Neumann boundary conditions; these and other details are described in [17].

3 Simulation Results for Gray Level Images

We ran a series of numerical simulations to demonstrate the performance of the implicit scheme for the Beltrami flow. We introduced an acceleration factor f that is defined as the ratio of the step size used in the implicit scheme to the maximum allowed step size for the explicit scheme. For a square grid, and assuming the pixels are a unit length apart, the maximun time step size for the explicit scheme is 0.25. We ran the scheme with values of f ranging between 1 and 200. The results are shown in Fig. 1. As we see, the implicit scheme is always stable, but for $f \gg 50$, the resulting accuracy may be insufficient for certain applications.

The next series of numerical simulations are carried out to study the edge enhancemeent effect on gray level images. We fix the acceleration factor to 10 and solve the following normalized reaction-diffusion equation in this series:

$$\frac{\partial U}{\partial t} \;=\; \cos\beta \;\; \nabla h \cdot \nabla U \;+\; \sin\beta \;\; h\, \nabla^2 U \tag{13}$$

The first term on the right side of Eq. (13) is a reaction term, while the second is a diffusion term and β is a parameter controlling the relative contribution of these opposing effects. The reaction term is responsible for edge enhancement, while the diffusion term smooths the noise away from the edges. Results of varying β between 0 to 90^o is shown in Fig. 2. The first row in Fig. 2, presents the initial image and the results for $\beta = 0$ (pure reaction) and $\beta = 30^o$. The second row presents the results for $\beta = 45^o$ (a nonlinear diffusion flow equation), $\beta = \arctan 2 \approx 63.4^o$ (the Beltrami flow), and $\beta = 90^o$ (scaled "linear" diffusion). According to Eq. (13), the edge enhancement effect should decay with increasing β. Indeed, we see that the edge enhancement is stronger for the nonlinear diffusion flow ($\beta = 45^o$) than for the Beltrami flow ($\beta = 63.4^o$).

Fig. 1. Results of implicit difference scheme for Beltrami flow till scale = 250; the first row shows the initial image and results with $f = 1, 2$; second row corresponds to $f = 5, 10, 20$; third row corresponds to values of $f = 50, 100, 200$.

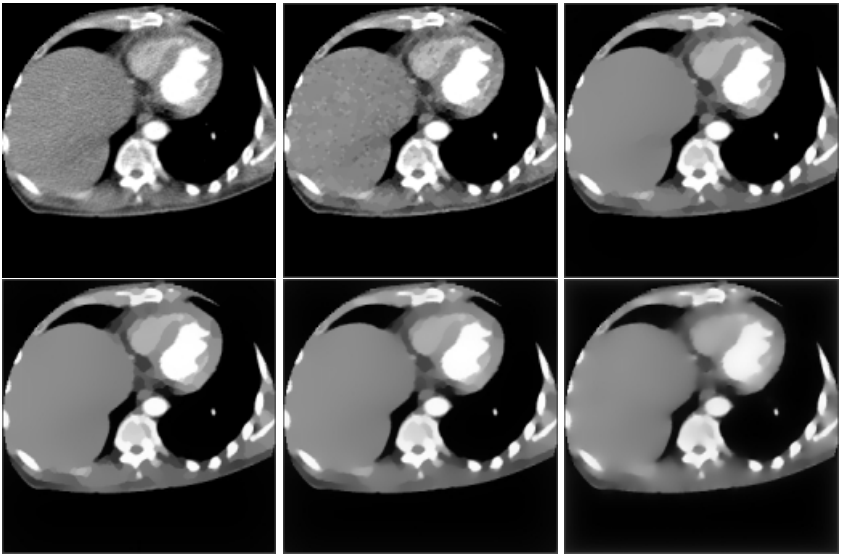

Fig. 2. Results of solving Eq. 13 until scale = 250 with different values of β.

In Fig. 3, the implicit difference scheme in $3D$ for the Beltrami flow is applied to a volumetric image of the brain. The image consists of 124 slices of 256×256 images. In the first column of Fig. 3, we show the original slice # 20, and results of running the flow until scale = 50 and 100. The other two columns show exactly the same results for slice #50 and #100. An acceleration factor of 12 was applied in all the simulations. We note that the $3D$ Beltrami flow can also be arranged as follows:

$$\dot{U} \;=\; \nabla \cdot \left(\frac{h \nabla U}{2} \right) \;+\; \frac{h \nabla^2 U}{2} \tag{14}$$

where $h = 1/(1 + U_x^2 + U_y^2 + U_z^2)$ is the edge indicator function.

Note that due to the additional component U_z^2 in the denominator, the $3D$ Beltrami flow is slower than the corresponding $2D$ flow. This means that to achieve the same degree of noise reduction and edge enhancement, larger scale values should be employed in $3D$.

4 Beltrami Smoothing for Color Images

The Beltrami flow for color images is governed by the following set of partial differential equations [11,14]:

$$\frac{\partial I_i}{\partial t} \;=\; \frac{\dfrac{\partial P_i}{\partial x} \;+\; \dfrac{\partial Q_i}{\partial y}}{g} \;-\; \frac{\dfrac{\partial g}{\partial x}\, P_i + \dfrac{\partial g}{\partial y}\, Q_i}{2\, g^2} \tag{15}$$

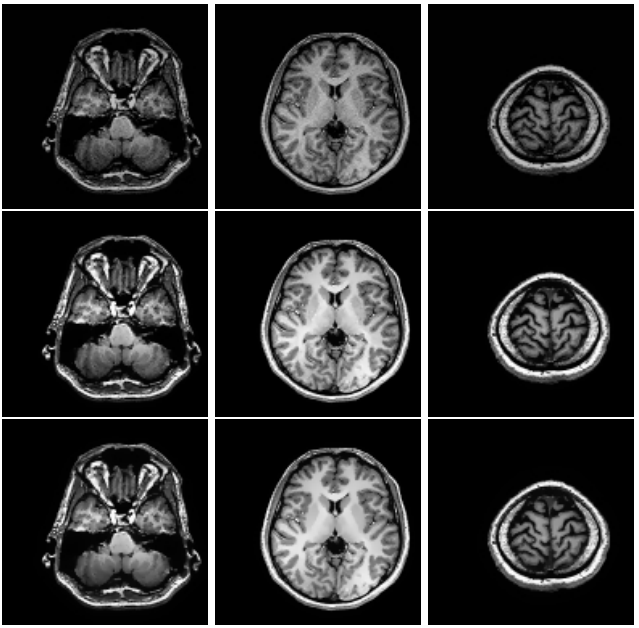

Fig. 3. Results of $3D$ Beltrami edge enhancing flow

where $i = 1, 2, 3$ corresponds to the color channel (red, green, blue), I_i is the corresponding pixel value, and P_i, Q_i are defined by:

$$P_i \;=\; g_{22}\,\frac{\partial I_i}{\partial x} \;-\; g_{12}\,\frac{\partial I_i}{\partial y} \qquad\qquad Q_i \;=\; -\,g_{12}\,\frac{\partial I_i}{\partial x} \;+\; g_{11}\,\frac{\partial I_i}{\partial y}. \tag{16}$$

g_{11}, g_{12} and g_{22} are components of a symmetric matrix (tensor) $\boldsymbol{G}$ of dimension 2×2, and g is its discriminant, i.e.

$$\boldsymbol{G} \;=\; \begin{bmatrix} g_{11} & g_{12} \\ g_{12} & g_{22} \end{bmatrix} \tag{17}$$

$$g_{11} \;=\; 1 \;+\; \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial x}\right)^2 \qquad g_{22} \;=\; 1 \;+\; \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial y}\right)^2$$

$$g_{12} \;=\; \sum_{j=1}^{3} \frac{\partial I_j}{\partial x}\,\frac{\partial I_j}{\partial y} \qquad\qquad g \;=\; \det \boldsymbol{G} \;=\; g_{11}\,g_{22} \;-\; g_{12}^2. \tag{18}$$

For the time being, instead of Beltrami flow (15), let us consider a simplified smoothing flow, which is an analog of the nonlinear diffusion for gray level image:

$$\frac{\partial I_i}{\partial t} \;=\; \frac{\dfrac{\partial P_i}{\partial x} \;+\; \dfrac{\partial Q_i}{\partial y}}{g} \;-\; \frac{\dfrac{\partial g}{\partial x}\,P_i + \dfrac{\partial g}{\partial y}\,Q_i}{g^2} \tag{19}$$

Note that the factor 2 is missing in the denominator of Eq. (19). This form is easier to handle in the following analysis. The above equation can be rearranged into a vector equation, namely

$$\frac{\partial I}{\partial t} \;=\; \frac{\partial}{\partial x}\left(\frac{P}{g}\right) \;+\; \frac{\partial}{\partial y}\left(\frac{Q}{g}\right) \tag{20}$$

Let us define vector $\boldsymbol{S}_i$ of length 2, whose components are P_i/g and Q_i/g. We can then show that

$$\boldsymbol{S}_i \;=\; \boldsymbol{G}^{-1} \,\nabla\, I_i \;=\; \boldsymbol{R} \,\nabla\, I_i \tag{21}$$

where $\boldsymbol{R}$ is the inverse of matrix $\boldsymbol{G}$, and the component wise flow becomes

$$\frac{\partial I_i}{\partial t} \;=\; \nabla \cdot \boldsymbol{S}_i \;=\; \nabla \cdot (\boldsymbol{R}\, \nabla I_i) \tag{22}$$

Note that there are three pixel value gradient vectors ∇I_i (for each color component), three vector components $\boldsymbol{S}_i$, but only one matrix $\boldsymbol{R}$. This is 2×2 edge indicator matrix, similar to a scalar edge indicator function for a gray level case. Expanding the last equation, we obtain a reaction term with the first derivatives of pixel values, and a diffusion term with containing the second derivatives

$$\frac{\partial I_i}{\partial t} \;=\; (\nabla \cdot \boldsymbol{R}) \cdot\; \nabla I_i \;+\; \boldsymbol{R} \cdot \nabla\nabla I_i. \tag{23}$$

The first term is a scalar product of two vectors (divergence of tensor $\nabla \cdot \boldsymbol{R}$ yields a vector), and $\boldsymbol{R} \cdot \nabla\nabla I_i$ is a full scalar product of two tensors; $\nabla\nabla I_i$ is the following matrix of second derivatives:

$$\nabla\nabla I_i \;=\; \begin{bmatrix} I^i_{xx} & I^i_{xy} \\ I^i_{xy} & I^i_{yy} \end{bmatrix} \tag{24}$$

Note that while this nonlinear diffusion flow includes the reaction term and the diffusion term, the Beltrami color flow includes an additional reaction term not considered here.

Now, consider the nonlinear diffusion flow, Eq. (22), and replace the action of matrix $\boldsymbol{R}$ on vector ∇I_i by the action of the eigenvalue on that vector, see [18,15, 12]. For this, we de-compose the gradient vector ∇I_i into the basis of principal directions $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$, as follows:

$$\nabla I_i \;=\; k_1 \boldsymbol{V}_1 \;+\; k_2 \boldsymbol{V}_2, \tag{25}$$

where $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ are normalized eigenvectors of the edge indicator tensor $\boldsymbol{R}$. Assume λ_1 and λ_2 are eigenvalues, corresponding to these eigenvectors, i.e.,

$$\boldsymbol{R}\boldsymbol{V}_k \;=\; \lambda_k \boldsymbol{V}_k \qquad\qquad k \;=\; 1,\; 2 \tag{26}$$

Consider the expression inside the brackets on the right side of Eq. (22),

$$\boldsymbol{R}\nabla I_i \;=\; \boldsymbol{R}(k_1 \boldsymbol{V}_1 \;+\; k_2 \boldsymbol{V}_2) \;=\; k_1 \boldsymbol{R}\boldsymbol{V}_1 + k_2 \boldsymbol{R}\boldsymbol{V}_2 \;=\; k_1 \lambda_1 \boldsymbol{V}_1 + k_2 \lambda_2 \boldsymbol{V}_2 \tag{27}$$

Note that λ_k are real positive numbers since matrix $\boldsymbol{R}$ is symmetric and positive definite. Recall that

$$g = 1+\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial x}\right)^2+\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial y}\right)^2+\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial x}\right)^2\cdot\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial y}\right)^2-\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial x}\frac{\partial I_j}{\partial y}\right)^2 \tag{28}$$

and the eigenvalues λ_k of matrix $\boldsymbol{R}$ are roots of a quadratic equation:

$$\lambda\cdot g = 1+\frac{\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial x}\right)^2+\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial y}\right)^2}{2}\pm\sqrt{\frac{\left[\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial x}\right)^2+\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial y}\right)^2\right]^2}{4}-A} \tag{29}$$

where

$$A \;=\; \sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial x}\right)^2\cdot\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial y}\right)^2 \;-\; \left(\sum_{j=1}^{3}\frac{\partial I_j}{\partial x}\frac{\partial I_j}{\partial y}\right)^2 \tag{30}$$

Expanding powers and products in Eq. (30) we can express A to be the sum of squares of lengths for cross-products of pairs of gradients $\nabla I_m \times \nabla I_{m+1}$;

$$A \;=\; \sum_{j=1}^{3}\left(\nabla I_j \times \nabla I_{(j+1)\ div\ 3}\right)^2 \tag{31}$$

For the gray level image, the edge is a line which divides region into segments with different pixel values. In other words, when passing across the edge, the pixel value changes in a discontinuous manner. The magnitude of the gradient vector is large and its direction is normal to the edge.

For the color image, the gradients of red-green-blue components may have different directions. However, we assume that in the proximity of the edge at least one of them has a large magnitude. The direction of this gradient is normal to the edge. The directions of other color gradients are collinear to this direction provided the magnitudes of gradients of these other components are also large. For the component(s) of small gradient magnitude, the direction of gradient does not matter. Thus, in the close proximity of the "true" edge, the value A in Eq. (31) will be small because all cross-products of gradients vanish or almost vanish; this is because either the components of the cross-product are collinear, or due to the fact that one of them or both have small magnitude.

Since the cross-products are small in the proximity of the edge, the value A in Eq. (29) may be neglected and the eigenvalues become:

$$\lambda_1 \;\approx\; \frac{1}{g} \qquad \lambda_2 \;\approx\; 1 \qquad g \;\approx\; 1 \;+\; \sum_{j=0}^{3}\left(\frac{\partial I_j}{\partial x}\right)^2 \;+\; \sum_{j=0}^{3}\left(\frac{\partial I_j}{\partial y}\right)^2 \tag{32}$$

In the proximity of the edge, the principal direction $\boldsymbol{V}_1$ of the edge indicator matrix $\boldsymbol{R}$, corresponds to the smaller eigenvalue $1/g$:

$$\frac{V_1^y}{V_1^x} \;=\; \sqrt{\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial y}\right)^2 \Bigg/ \sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial x}\right)^2} \tag{33}$$

Since we assumed that all gradients have approximately collinear directions in the proximity of the edge, it follows from Eq. (33) that the principal direction $\boldsymbol{V}_1$ coincides with these gradients and is normal to the edge. The second principal direction, corresponds to a larger eigenvalue 1:

$$\frac{V_2^x}{V_2^y} \;=\; -\sqrt{\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial x}\right)^2 \Bigg/ \sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial y}\right)^2} \tag{34}$$

The second eigenvector is tangent to the edge. Note that even when the cross products of color components' gradients do not vanish exactly, it is reasonable to define the direction of the edge as a principal direction of the edge indicator matrix corresponding to the larger eigenvalue $\lambda_2 \approx 1$.

Now consider Eq. (27). Since the edge is normal to the gradient,

$$k_1 \;\approx\; |\nabla I_i| \qquad\qquad k_2 \;\approx\; 0 \tag{35}$$

$$\boldsymbol{R}\nabla I_i \;=\; k_1\lambda_1\boldsymbol{V}_1 \;+\; k_2\lambda_2\boldsymbol{V}_2 \;\approx\; \frac{\nabla I_i}{g} \tag{36}$$

where g is given by Eq. (32). In the proximity of the edge, the governing equation for the color image flow becomes:

$$\frac{\partial I_i}{\partial t} \;=\; \nabla\cdot(\boldsymbol{R}\nabla I_i) \;=\; \nabla\cdot\left(\frac{\nabla I_i}{g}\right) \;=\; \nabla\left(\frac{1}{g}\right)\cdot\nabla I_i \;+\; \frac{\nabla^2 I_i}{g} \tag{37}$$

Now, recall that Eq. (37) describes the nonlinear diffusion flow which differs from the Beltrami flow. It follows from Eqs. (15, 19 and 37) that the Beltrami flow has an additional reaction term, namely

$$\frac{\partial I_i}{\partial t} \;=\; \nabla\left(\frac{1}{g}\right)\cdot\nabla I_i \;+\; \frac{\nabla^2 I_i}{g} \;+\; \frac{\dfrac{\partial g}{\partial x}\,P_i + \dfrac{\partial g}{\partial y}\,Q_i}{2\,g^2}. \tag{38}$$

Rearranging the last term in Eq. (38) to a tensor form:

$$\frac{\dfrac{\partial g}{\partial x}\,P_i + \dfrac{\partial g}{\partial y}\,Q_i}{2\,g^2} \;=\; -\,P_i\,\frac{\partial}{\partial x}\,\frac{1}{2\,g} \;-\; Q_i\,\frac{\partial}{\partial y}\,\frac{1}{2\,g} \;=\; -\,\nabla\left(\frac{1}{2\,g}\right)\cdot g\;\boldsymbol{R}\;\nabla I_i \tag{39}$$

and applying Eq. (36) in the proximity of the edge, i.e., $g\ \boldsymbol{R}\ \nabla I_i\ \approx\ \nabla I_i$, the governing equation (38) for Beltrami flow can be expressed as:

$$\frac{\partial I_i}{\partial t}\ \approx\ \nabla\left(\ \frac{1}{2\ g}\ \right)\cdot\nabla I_i\ +\ \frac{\nabla^2 I_i}{g}\ =\ \nabla\cdot\left(\frac{\nabla I_i}{2\ g}\right)\ +\ \frac{\nabla^2 I_i}{2\ g} \tag{40}$$

This corresponds exactly with the governing partial differential equation for the Beltrami flow equation fro a gray level image. This means that the mechanism of the edge enhancement is exactly the same. In the proximity of the edge, $1/g$ reaches minimum values. The gradient of $1/g$ is directed outside the thin pass of the edge. The gradient of the pixel value is also normal to the edge, but its direction coincides with the gradient of $1/g$ for larger pixel values, and is opposite to that direction for smaller pixel values. The reactive component of the equation becomes positive for larger pixel values and negative for smaller pixel in the proximity of the edge. So, the large values become even larger, and the small values become even smaller. thereby enhancing and sharpening the edge.

Note that the simplified decoupled form of the Beltrami smoothing may be used for numerical computations. Close to the edge this form is justified because the coupling between x and y components of gradient becomes weak. Away from the edge, the decoupled form brings a definite inaccuracy, and the flow is no longer exactly Beltrami. However, we consider that the accuracy is crucial at the proximity of the edge and less important elsewhere. In this case, the decoupled form of the governing equation (40) leads to a considerable saving of the computational time. Furthermore, the decoupled form makes it possible to apply the additive splitting algorithm leading to a semi-implicit linearized difference scheme, and this yields much faster and unconditionally stable computation. For a specific raw or column of pixels, the left-side matrix of the difference equation set is the same for all three color components, and only the right-side vectors differ. Thus, we solve the three sets of equations simultaneously.

5 Simulation Results for Color Images

The goal of this numerical experiment is to show that the weakly coupled Beltrami smoothing operator may be replaced by its decoupled approximation without essential loss of accuracy and with a great saving of the computational time.

Fig. 4 presents the simulation results for implicit scheme with different values of acceleration factor $f\ =\ 2\ldots 50$. The first row includes the initial image (left picture) and the smoothing simulation results for the Beltrami filtering, using the coupled explicit difference scheme with a full edge indicator matrix (central picture) and explicit difference scheme with decoupled governing equation with the eigenvalue instead of the mathix (right picture). In the second row the results are plotted for implicit difference scheme with decoupled governing equation and acceleration factor $f\ =\ 1,\ 2,\ 5$. The third row corresponds to $f\ =\ 10,\ 20,\ 50$. As we see, for acceleration factors of up to 20, the flow is simulated with a reasonable accuracy.

Fig. 4. Semi-implicit AOS implementation of Beltrami flow for Color Images; scale 1000

6 Closing Remarks

The anisotropic Beltrami operator for gray level images is reduced to reaction-diffusion form and this makes it possible to apply the Additive Operator Split (AOS) approach. Based on this approach, the uncondiotionally stable difference scheme is developed. The method uses the known values of the edge indicator function from the previous scale step, and incorporates the unknown pixel values from the next step, thus making the difference scheme semi-implicit and linearized. The implicit scheme leads to considerable saving of the computational time as compared to the explicit scheme: up to ten times and even more, depending on the value of the scale step, with no visible loss of accuracy. The approach may be applied also to the mean curvature flow in a similar way, with a different edge indicator function.

The eigenvalue analysis is applied to study the Beltrami smoothing technique for color images. It is shown that in the proximity of the edge the coupling of the Beltrami operator becomes weak. The principal directions of the edge indicator matrix are normal to the edge and tangent to the edge. This allows us to replace the action of the edge indicator matrix on the gradient vector by the action of its eigenvalue on that vector. The main assumption for fast processing of color images is that the cross-products of the gradients of color components are negligibly small in the proximity of edge, since their directions are all parallel and normal to the edge. When these cross-products are small, the coupling becomes weak. Weak coupling of the Beltrami operator for color images makes it possible to apply the AOS technique and unconditionably stable fast implicit difference scheme. The difference scheme is similar to that for gray-level images. Numerical simulations confirm the validity of the assumption about the weak coupling.

References

1. P. Perona, J. Malik, "Scale-space and edge-detection using anisotropic diffusion," IEEE Trans. of PAMI, Vol. 12, pp. 629–639, 1990.
2. L. Alvarez, P.L. Lions, J.M. Morel, "Image selective smoothing and edge detection by non-linear diffusion II," SIAM Journal on Numerical Analysis, Vol. 29(3), pp. 845–866, 1992.
3. L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," Physica D, Vol. 60, pp. 259–268, 1992.
4. J. Weickert, B.M. ter Haar Romeny, and M.A. Viergever. "Efficient and reliable scheme for nonlinear diffusion filtering". *IEEE Trans on Image Processing.*, Vol. 7(3), pp. 398-410 (1998).
5. R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. "Fast Geodesic Active Contours". M. Nielsen, P. Johansen, O.F. Olsen, J. Weickert (Editors), Scale-space theories in computer vision, Lecture Notes in Computer Science, Vol. 1682, Springer, Berlin, 1999.
6. S. Osher and J. A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation," Journal of Computational Physics, Vol. 79, pp. 12-49, 1988.

7. R. Malladi and J. A. Sethian, "Image processing via level set curvature flow," Proc. of Natl. Acad. of Scie., Vol. 92, pp. 7046–7050, July 1995.
8. R. Malladi and J. A. Sethian, "Image Processing: Flows under Min/Max curvature and Mean curvature," Graphical Models and Image Processing, Vol. 58(2), pp. 127–141, 1996.
9. G.I. Barenblatt. "Self-Similar Intermediate Asymptotics for Nonlinear Degenerate Parabolic Free-Boundary Problems which Occur in Image Processing". to appear in *Proceedings of the National Academy of Science*, 2001.
10. B. M. ter Harr Romeny, editor, Geometry-driven diffusion in computer vision, Kluwer Academic publishers, The Netherlands, 1994.
11. R. Kimmel, R. Malladi, and N. Sochen. "Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images". *International Joornal of Computer Vision*, 1999.
12. R. Kimmel, "A natural norm for color processing," in *Proceedings of 3-rd ACCV*, Springer-Verlag LNCS 1352, pp. 88–95, Hong-Kong, January 1998.
13. N. Sochen, R. Kimmel and R. Malladi, "From high energy physics to low level vision," LBNL Report # 39243, Lawrence Berkeley National Laboratory, University of California, Berkeley, Aug. 1996.
14. N. Sochen, R. Kimmel and R. Malladi, "A genearal framework for low level vision". *IEEE Transactions*, Vol. 7, No. 3, 1998.
15. G. Sapiro, "Vector-valued active contours," in *Proceedings of CVPR '96*, pp. 650–655, 1996.
16. A. M. Polyakov, "Quantum geometry of bosonic strings," in *Physics Letters B*, 103B(3), pp. 207–210, 1981.
17. R. Malladi and I. Ravve, "Fast difference scheme for anisotropic Beltrami smoothing and edge contrast enhancement of gray level and color images," LBNL report # 48796, Lawrence Berkeley National Laboratory, University of California, Berkeley, August, 2001.
18. J. Weickert. *Anisotropic Diffusion in Image Processing*. Ph. D. Thesis, Kaiserslautern University, 1996.
19. Curtis E. Gerald and Patric O. Wheatley. *Applied Numerical Analysis*, Addison Wesley, NY, 1999.
20. William H. Press, Saul A. Tekolsky, William T. Vetterling and Brian F. Flannery. *Numerical Recipes in C*, Cambridge University Press, 1992
21. A. Yezzi, "Modified curvature motion for image smoothing and enhancement," IEEE Tran. of Image Processing, Vol. 7, No. 3, 1998.