LBNL-48796
Preprint

ERNEST ORLANDO LAWRENCE
BERKELEY NATIONAL LABORATORY

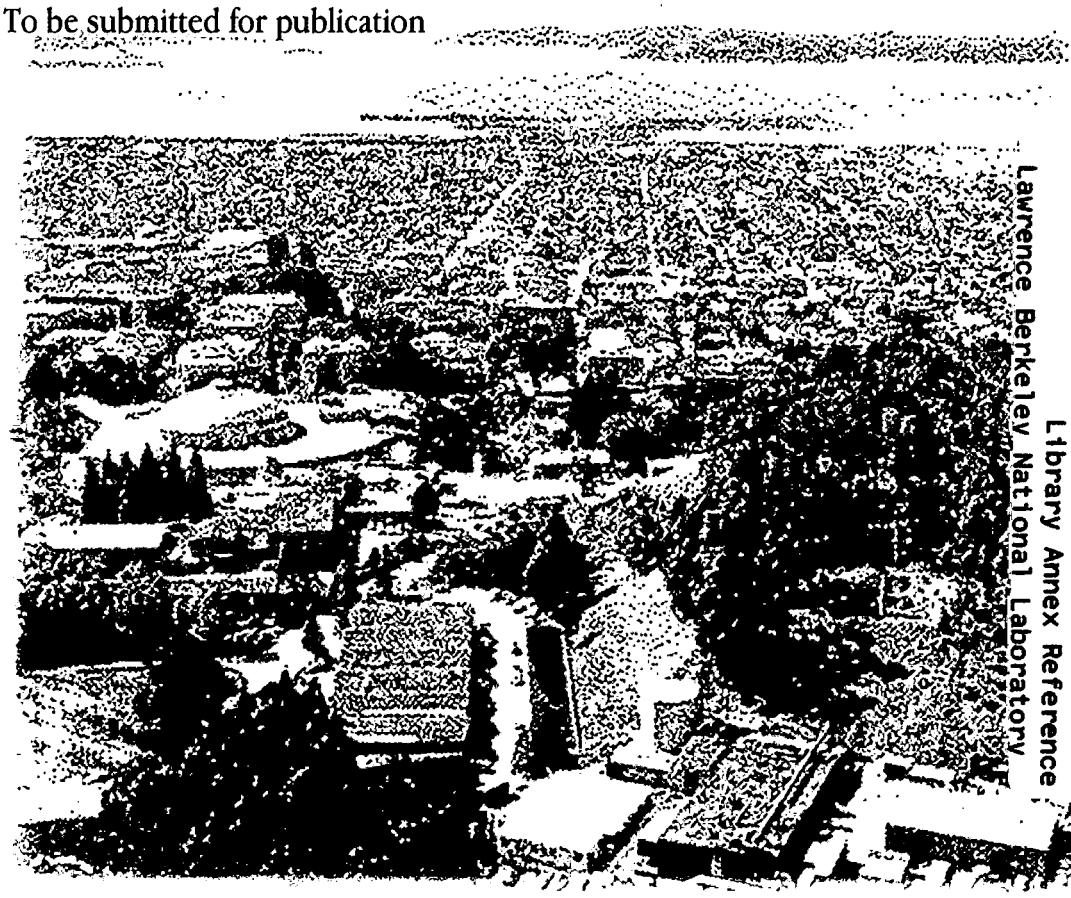
Fast Difference Scheme for Anisotropic Beltrami Smoothing and Edge Contrast Enhancement of Gray Level and Color Images

R. Malladi and I. Ravve

Computing Sciences Directorate
Mathematics Department

August 2001

To be submitted for publication



DISCLAIMER

LBNL-48796

Fast Difference Scheme for Anisotropic Beltrami Smoothing and Edge Contrast Enhancement of Gray Level and Color Images

R. Malladi and I. Ravve

Computing Sciences Directorate
Ernest Orlando Lawrence Berkeley National Laboratory
University of California
Berkeley, California 94720

August 2001

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

Fast Difference Scheme for Anisotropic Beltrami Smoothing and Edge Contrast Enhancement of Gray Level and Color Images

R. Malladi and I. Ravve

Lawrence Berkeley National Laboratory

Abstract

In this study, we re-arrange the governing equation for Beltrami diffusion flow to the form, where the reaction term and the diffusion term appear explicitly. The reaction term leads to an edge enhancement, and we illustrate this phenomenon by numerical simulations. The reaction-diffusion form of the Beltrami equation makes it possible to develop the unconditionally stable semi-implicit scheme for image filtering. The method is based on Additive Operator Split, applied originally by Weickert for the nonlinear diffusion flow. 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 Beltrami flow. This approach leads to a semi-implicit linearized difference scheme. The sources of truncation error are examined and its value is established. The computational time required for image filtering may be saved up to ten times or even more, depending on the value of the scale step. A similar approach may be applied also for mean-curvature flow.

For the color images, the edge indicator function is presented by a matrix. Equations for two-dimensional images become coupled, and this does not allow to apply the AOS splitting immediately. However, in the proximity of the edge, the cross-products of gradients for color

components all vanish or almost vanish. The Beltrami smoothing operator becomes weakly coupled. The principal directions of the edge indicator matrix are normal to the edge and tangent to the edge. Replacing the action of this matrix on the gradient vector by an action of its eigenvalue, we reduce the color problem to the gray level case and show that the scalar edge indicator function for the color case is exactly the same as that for the gray level image. Thus, the interprertation of the gray level edge enhancement mechanism caused by reaction term of the smoothing PDE may be applied for color images, and the fast implicit smoothing scheme may be implemented for color images as well.

1 Introduction

Smoothing of noisy images presents usually a numerical integration of a parabolic PDE in scale and two dimensions in space. This is often the most time consumptive component of image processing algorithms. Smoothing technique governed by the Beltrami flow (7) proves to be the most effective as:

- the Beltrami equation is normalized by the gradient length in power 4
- the Beltrami flow incorporates the edge indicator function, thus providing a minimum diffusion at the edges and extensive diffusion elsewhere (inside and outside the particles)

However, on the other hand, the normalization makes the Beltrami smoothing slow. The explicit numerical integration scheme is conditionally stable. It imposes a constraint on the time step length, depending on the spatial resolution $s = \Delta x = \Delta y$:

$$\Delta t \ \leq \ ks^2 \tag{1}$$

It proves to be difficult to establish analytically the limiting value of k for the nonlinear smmothing. For the linear smoothing governed by a parabolic PDE

$$\dot{U} \ = \ \frac{\partial^2 U}{\partial x^2} \ + \ \frac{\partial^2 U}{\partial y^2}, \tag{2}$$

the ultimate time step value is:

$$\Delta t_{\max} = \frac{1}{2}\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)^{-1} \tag{3}$$

and thus, for an equal grid in x and y, $k = 1/4$ in Eq. (1). The linear parabolic PDE (2) can be solved semi-analytically, applying the Fast Fourier Transform. It can be also solved numerically, applying the ADI technique (Alternate Directions Implicit) where the Laplacian is approximately split into a product of two operators, in x only and y only. Both methods lead to a fast computational procedure. However, the linear smoothing has a number of disadvantages; in paricular, the governing linear PDE (2) does not refer to the edge indicator function and the gradient length and thus can not enhance the edge (moreover, the edge may bes damped out). Therefore, for images with moderarte and strong noise, a nonlinear smoothing is usually applied.

Theoretical and numerical aspects of the nonlinear anisotropic diffusion were extensively studied by Weickert [6]. This type of smoothing should beapplied in cases when the edge sharpening is required . The formation of sharp steps at the edges of images was studied by G. I. Barenblatt [3]. In this work, an asymptotic self-similar solution was obtained for a particular case of the Beltrami equation, leading to edge enhancement.

It proves to be difficult to define analytically the threshold value of k for nonlinear smoothing, but various numerical tests revealed that for Beltrami smoothing (7), the threshold vaslue is almost exactly the same as for the linear diffusion (2): $k \approx 1/4$. Thus, the unconditionally stable numerical scheme becomes an important matter. In this case, the time step will be limited only by the accuracy of the solution, and not by the stability issues. We will further show that for image smoothing, the accuracy factor is not crucial, but the stability threshold is usually a bottleneck.

The Additive Operator Splitting (AOS) Schemes were introduced by Weickert et al. [1] as unconditionally stable for the nonlinear diffusion in image processing and then applied by Goldenberg et al. [2] to implement a fast version of the geodesic contour model. In this Report, we apply the AOS

technique to the Beltrami flow, which differs from the nonlinear diffusion, but may be treated in a similar way. Applying a different edge indicator, the AOS can be used also for the Curvature flow.

2 Splitting the Beltrami Operator

Compare the nonlinear diffusion of a gray-level image with Beltrami flow. Let U be the pixel value. The nonlinear diffusion is described by the following PDE [1, 2]

$$\dot{U} \;=\; \nabla \cdot \left(\frac{\nabla U}{g} \right) \;=\; \frac{\partial}{\partial x} \frac{U_x}{g} \;+\; \frac{\partial}{\partial y} \frac{U_y}{g} \tag{4}$$

where $h \;=\; \dfrac{1}{g}$ is the Edge Indicator Function and

$$g \;=\; U_x^2 \;+\; U_y^2 \;+\; 1 \tag{5}$$

Introducing (5) into (4), we obtain:

$$\dot{U} \;=\; \frac{U_{xx} \left(-U_x^2 + U_y^2 + 1 \right) \;-\; 4\, U_{xy} U_x U_y \;+\; U_{yy} \left(U_x^2 - U_y^2 + 1 \right)}{\left(U_x^2 + U_y^2 + 1 \right)^2} \tag{6}$$

Now consider the Beltrami flow:

$$\begin{aligned}
\dot{U} \;&=\; \frac{U_{xx} \left(U_y^2 + 1 \right) \;-\; 2 U_{xy} U_x U_y \;+\; U_{yy} \left(U_x^2 + 1 \right)}{\left(U_x^2 + U_y^2 + 1 \right)^2} \\
&= \frac{\partial}{\partial x} \frac{U_x}{g} + \frac{\partial}{\partial y} \frac{U_y}{g} + \frac{U_x U_{xx} + U_y U_{xy}}{\left(U_x^2 + U_y^2 + 1 \right)^2} \cdot U_x + \frac{U_x U_{xy} + U_y U_{yy}}{\left(U_x^2 + U_y^2 + 1 \right)^2} \cdot U_y \\
&= \frac{\partial}{\partial x} \frac{U_x}{g} + \frac{\partial}{\partial y} \frac{U_y}{g} - \frac{g_x U_x}{2} - \frac{g_y U_y}{2} \\
&= \frac{1}{2} \frac{\partial}{\partial x} \frac{U_x}{g} + \frac{1}{2} \frac{\partial}{\partial y} \frac{U_x}{g} + \frac{1}{2\, g} \left(U_{xx} + U_{yy} \right) \;=\; \nabla \cdot \left(\frac{\nabla U}{2\, g} \right) \;+\; \frac{\nabla^2 U}{2\, g}
\end{aligned} \tag{7}$$

Eq. (7) presents Beltrami flow with the mixed derivative U_{xy} hidden. This makes it possible to apply Additive Operator Split, to be used in the implicit numerical scheme for individual rows and columns of pixels:

$$\dot{U} \;=\; (\; A_x \;+\; A_y \;)\; U \tag{8}$$

where A_x and A_y are differential operators:

$$\begin{cases} A_x \;=\; \dfrac{\partial}{\partial x}\left(\dfrac{1}{2\,g}\,\dfrac{\partial}{\partial x}\right) \;+\; \dfrac{1}{2\,g}\,\dfrac{\partial^2}{\partial x^2} \\[2ex] A_y \;=\; \dfrac{\partial}{\partial y}\left(\dfrac{1}{2\,g}\,\dfrac{\partial}{\partial y}\right) \;+\; \dfrac{1}{2\,g}\,\dfrac{\partial^2}{\partial y^2} \end{cases} \tag{9}$$

3 Splitting the Curvature Flow

For a different edge indicator function $\dfrac{1}{g}$ and

$$g \;=\; \sqrt{U_x^2 \;+\; U_y^2 \;+\; 1}, \tag{10}$$

the splitting technique can be applied for the image smoothing with the Curvature flow:

$$\begin{aligned} \dot{U} \;&=\; \frac{U_{xx}\left(U_y^2+1\right) \;-\; 2U_{xy}U_xU_y \;+\; U_{yy}\left(U_x^2+1\right)}{U_x^2+U_y^2+1} \\ &=\; g\left(\frac{\partial}{\partial x}\,\frac{U_x}{g} \;+\; \frac{\partial}{\partial y}\,\frac{U_y}{g}\right) \end{aligned} \tag{11}$$

4 Semi-Implicit Scheme for Beltami Flow

Apply the backward difference to Eq. (8).

$$\frac{\boldsymbol{U}^{n+1} \;-\; \boldsymbol{U}^n}{\Delta t} \;=\; (\boldsymbol{A}_x \;+\; \boldsymbol{A}_y)\; \boldsymbol{U}^{n+1} \tag{12}$$

The superscript $n+1$ is related to the next time step, and n - to the present. The subscripts i, j index the location of the pixel. $U_{i,j}^n$ are given values, and

$U_{i,j}^{n+1}$ are to be found. Applying U^{n+1} on the right side of Eq. (12) makes the integration scheme implicit and unconditionally stable:

$$[\boldsymbol{I} - \Delta t\,(\boldsymbol{A}_x + \boldsymbol{A}_y)\,]\;\boldsymbol{U}^{n+1} \;=\; \boldsymbol{U}^n \tag{13}$$

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 actually on the gradient of U, we treat it like a given function of (x, y), and the nonlinear governing PDE becomes "quasi-linear".

Eq. (13) includes a large bandwidth matrix, because all equations, related to new pixel values $\boldsymbol{U}^{n+1}$ are coupled. Our aim is to decouple set (13) so that each row and each column of pixels will be treated individually. For this, we re-arrange the resolving set:

$$\boldsymbol{U}^{n+1} \;=\; [\boldsymbol{I} - \Delta t\,(\boldsymbol{A}_x + \boldsymbol{A}_y)\,]^{-1}\;\boldsymbol{U}^n \tag{14}$$

Of course, we are not going to inverse the matrix to solve the linear set. This is only a symbolic form used forv further derivation. For a small value of Δt, the matrix in the brackets on the right side of Eq. (14) is close to the identity $\boldsymbol{I}$. Thus, its inverse can be expanded into the Taylor series at the proximity of $\boldsymbol{I}$:

$$[\boldsymbol{I} - \Delta t\,(\boldsymbol{A}_x + \boldsymbol{A}_y)\,]^{-1} \;\approx\; \boldsymbol{I} + \Delta t\,(\boldsymbol{A}_x + \boldsymbol{A}_y) \tag{15}$$

where only the linear term are retained and the high order terms are neglected. Introducing Eq. (15) into (14), we get:

$$2\boldsymbol{U}^{n+1} \;=\; (\boldsymbol{I} + 2\Delta t\boldsymbol{A}_x)\,\boldsymbol{U}_n \;+\; (\boldsymbol{I} + 2\Delta t\boldsymbol{A}_y)\,\boldsymbol{U}^n \tag{16}$$

Introduce the new notations $\boldsymbol{V}$ and $\boldsymbol{W}$:

$$(\boldsymbol{I} + 2\Delta t\boldsymbol{A}_x)\,\boldsymbol{U}^n \;=\; \boldsymbol{V} \qquad\qquad (\boldsymbol{I} + 2\Delta t\boldsymbol{A}_y)\,\boldsymbol{U}^n \;=\; \boldsymbol{W} \tag{17}$$

The solution includes now two components:

$$\boldsymbol{U}^{n+1} \;=\; \frac{\boldsymbol{V} + \boldsymbol{W}}{2} \tag{18}$$

We like to get an implicit scheme, so we re-arrange Eq. (17) to apply the differential matrix operators $\boldsymbol{A}_x$ and $\boldsymbol{A}_y$ to $\boldsymbol{U}^{n+1}$ (and not to $\boldsymbol{U}^n$).

$$(\boldsymbol{I} + 2\Delta t \boldsymbol{A}_x)^{-1} \boldsymbol{V} = \boldsymbol{U}^n \qquad (\boldsymbol{I} + 2\Delta t \boldsymbol{A}_y)^{-1} \boldsymbol{W} = \boldsymbol{U}^n \tag{19}$$

Expanding the inverse matrices into the Taylor series and applying the linearization for small Δt, we finally obtain the equation sets for $\boldsymbol{V}$ and $\boldsymbol{W}$:

$$(\boldsymbol{I} - 2\Delta t \boldsymbol{A}_x) \boldsymbol{V} = \boldsymbol{U}^n \qquad (\boldsymbol{I} - 2\Delta t \boldsymbol{A}_y) \boldsymbol{W} = \boldsymbol{U}^n \tag{20}$$

5 Finite Difference Equation

The differential operators A_x and A_y in Eq. (9) are similar, and therefore we derive here the difference equation the difference equation for a single row of pixels. Equation for the column of pixels is identical. Consider a row with $N + 1$ pixels enumerated from 0 to N, Fig. 1.

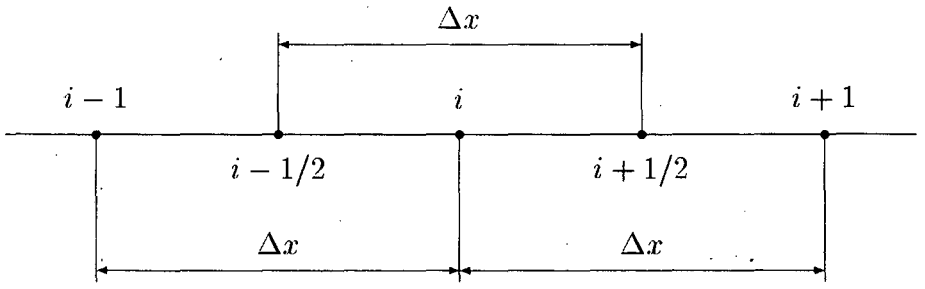


Figure 1: Finite Difference Scheme

$$2\, A_x\, V = \frac{\partial}{\partial x} \left(\frac{1}{g} \frac{\partial V}{\partial x} \right) + \frac{1}{g} \frac{\partial^2 V}{\partial x^2} \tag{21}$$

In the difference equations (22 - 24) the omitted error terms are of order $O(\Delta x^2)$.

$$\left(\frac{\partial V}{\partial x} \right)_{i+1/2} = \frac{V_{i+1} - V_i}{\Delta x}; \qquad \left(\frac{\partial V}{\partial x} \right)_{i-1/2} = \frac{V_i - V_{i-1}}{\Delta x} \tag{22}$$

$$\left(\frac{\partial^2 V}{\partial x^2} \right)_i = \frac{V_{i+1} - 2V_i + V_{i-1}}{\Delta x^2} \tag{23}$$

$$2\left(A_x V\right)_i = \frac{h_{i+1/2}\left(V_{i+1} - V_i\right) - h_{i-1/2}\left(V_i - V_{i-1}\right)}{\Delta_x^2} + h_i \frac{V_{i+1} - 2V_i + V_{i-1}}{\Delta x^2} \tag{24}$$

$$2\left(A_x V\right)_i = \frac{h_{i-1/2} + h_i}{\Delta x^2} V_{i-1} - \frac{h_{i-1/2} + 2h_i + h_{i+1/2}}{\Delta x^2} V_i + \frac{h_i + h_{i+1/2}}{\Delta x^2} V_{i+1} \tag{25}$$

To avoid establishing the values of the edge indicator function h at the non-nodal points $i - 1/2$ and $i + 1/2$, we average the values of two neighbour nodes:

$$h_{i+1/2} = \frac{h_i + h_{i+1}}{2} + O\left(\Delta x^2\right) \qquad h_{i-1/2} = \frac{g_{i-1} + h_i}{2} + O\left(\Delta x^2\right) \tag{26}$$

Introduce Eq. (26) into (25):

$$2\left(A_x V\right)_i = \frac{h_{i-1} + 3h_i}{2\Delta x^2} V_{i-1} - \frac{h_{i-1} + 6h_i + h_{i+1}}{2\Delta x^2} V_i + \frac{3h_i + h_{i+1}}{2\Delta x^2} V_{i+1} \tag{27}$$

It may seem that after introduction of Eq. (26) into (25), the overall error becomes huge, of order $O(1)$, due to Δ_x^2 in the denominator. However, due to a special symmetry of Eq. (27), the total error is still of order $O\left(\Delta_x^2\right)$. For proof we expand the functions $g(x)$ and $V(x)$ into the Taylor series in the proximity of point i and find the local truncation error of the difference operator $2A_x V$ with respect to the same differential operator. Let E_x^H be the local error for the right side of Eq. (25), and E_x^I - the same for Eq. (27), where superscript I stands for *integer* and H for *half-integer*.

$$\begin{aligned} E_x^H &= \left(\frac{\partial^3 h}{\partial x^3} \frac{\partial V}{\partial x} + 3 \frac{\partial^2 h}{\partial x^2} \frac{\partial^2 V}{\partial x^2} + 4 \frac{\partial h}{\partial x} \frac{\partial^3 V}{\partial x^3} + 4\, h \frac{\partial^4 V}{\partial x^4} \right) \frac{\Delta x^2}{24} \\ E_x^I &= \left(2 \frac{\partial^3 h}{\partial x^3} \frac{\partial V}{\partial x} + 3 \frac{\partial^2 h}{\partial x^2} \frac{\partial^2 V}{\partial x^2} + 2 \frac{\partial h}{\partial x} \frac{\partial^3 V}{\partial x^3} + 2\, h \frac{\partial^4 V}{\partial x^4} \right) \frac{\Delta x^2}{12} \end{aligned} \tag{28}$$

As we see, the erors are of the same order, and this justifies the validity of substitution (26). It follows from Eqs. (20 and 27)

$$-\Delta t\frac{h_{i-1}+3h_i}{2\Delta_x^2}V_{i-1}+\left(1+\Delta t\frac{h_{i-1}+6h_i+h_{i+1}}{2\Delta_x^2}\right)V_i-\Delta t\frac{3h_i+h_{i+1}}{2\Delta_x^2}V_{i+1}=U_i^n \tag{29}$$

Introduce the following notations:

$$\alpha_x = \frac{\Delta t}{2\Delta_x^2} \qquad \alpha_y = \frac{\Delta t}{2\Delta_y^2} \tag{30}$$

The finite difference equation comes to:

$$\begin{aligned} &-\alpha_x \ (\ h_{i-1} + 3h_i\ )\ V_{i-1} + [\ 1 + \alpha_x\ (\ h_{i-1} + 6h_i + h_{i+1}\ )\ ]\ V_i \\ &-\alpha_x\ (\ 3h_i + h_{i+1}\ )\ V_{i+1} = U_i^n \end{aligned} \tag{31}$$

Eq. (31) is a linear set with the three-diagonal matrix. The similar equation holds for W in y dimension, where $j = 0,\ 1,\ \ldots\ M$.

$$\begin{aligned} &-\alpha_y \ (\ h_{j-1} + 3h_j\ )\ W_{j-1} + [\ 1 + \alpha_y\ (\ h_{j-1} + 6h_j + h_{j+1}\ )\ ]\ W_j \\ &-\alpha_y\ (\ 3h_j + h_{j+1}\ )\ W_{j+1} = U_i^n \end{aligned} \tag{32}$$

For the first and the last nodes, either the Dirichlet, or the Neumann boundary conditions hold:

- In case of Dirichlet BC

$$V_0 = B; \qquad V_N = B; \qquad W_0 = B; \qquad W_M = B \tag{33}$$

  where B is a known value. We use central differences to establish the gradient components for the edge indicator function for all internal nodes of the grid and unilateral (forward or backward) differences at the boundary nodes, to find the normal derivarive of the pixel function U_n.

- In case of Neumann BC we assume that the normal derivatives V_n and W_n vanish along the rectangular contour of the computational box, i.e. we accept the mirror boundary conditions:

$$V_{-1} = V_1 \qquad \text{and} \qquad V_{N+1} = V_N \tag{34}$$

Establish the normal derivatives of the edge indicator function along the vertical and horizontal boundary lines:

$$\begin{cases} \dfrac{\partial h}{\partial x} &= -2\ \dfrac{U_x U_{xx} + U_y U_{xy}}{\left(1 + U_x^2 + U_y^2\right)^2} \\ \dfrac{\partial h}{\partial y} &= -2\ \dfrac{U_x U_{xy} + U_y U_{yy}}{\left(1 + U_x^2 + U_y^2\right)^2} \end{cases} \tag{35}$$

Along the vertical boundary $x = \text{const}$, $U_x = 0$ and $U_{xy} = 0$. Along the horizontal boundary $y = \text{const}$, $U_y = 0$ and $U_{xy} = 0$. Thus, in both cases $h_n = 0$, and the mirror boundary condition hold not only for the pixel value, but also for the edge indicator function. The ghost values become:

$$h_{-1} = h_1 \qquad \text{and} \qquad h_{N+1} = h_{N-1} \tag{36}$$

This affects the first and the last equations of Set (31):

$$\begin{aligned} [1 + \alpha_x (6h_0 + 2h_1)]\, V_0 \ &- \ \alpha_x (6h_0 + 2h_1)\, V_1 &&= U_1^n \\ -\alpha_x (2h_{N-1} + 6h_N)\, V_{N-1} \ &+ \ [1 + \alpha_x (2h_{N-1} + 6h_N)]\, V_N &&= U_N^n \end{aligned} \tag{37}$$

6 Local Truncation Eror

Let $\tilde{A}_x$ and $\tilde{A}_y$ be the difference operators (Eq. 27) that correspond to the differential operators A_x and A_y (Eq. 9). Then it follows from Eq. (28):

$$\tilde{A}_x\, U = A_x\, U + L_x\, U\, \frac{\Delta x^2}{24} \qquad \tilde{A}_y\, U = A_y\, U + L_y\, U\, \frac{\Delta y^2}{24} \tag{38}$$

where

$$\begin{aligned} L_x\, U &= 2\, \frac{\partial^3 h}{\partial x^3}\, \frac{\partial U}{\partial x} + 3\, \frac{\partial^2 h}{\partial x^2}\, \frac{\partial^2 U}{\partial x^2} + 2\, \frac{\partial h}{\partial x}\, \frac{\partial^3 U}{\partial x^3} + 2\, h\, \frac{\partial^4 U}{\partial x^4} \\ L_y\, U &= 2\, \frac{\partial^3 h}{\partial y^3}\, \frac{\partial U}{\partial y} + 3\, \frac{\partial^2 h}{\partial y^2}\, \frac{\partial^2 U}{\partial y^2} + 2\, \frac{\partial h}{\partial y}\, \frac{\partial^3 U}{\partial y^3} + 2\, h\, \frac{\partial^4 U}{\partial y^4} \end{aligned} \tag{39}$$

Rearrange Eq. (20) and expand into the Tailor series for small values of Δt:

$$\begin{aligned} V &= \left(1 - 2\Delta_t \tilde{A}_x\right)^{-1} U^n = U^n + 2\Delta t \tilde{A}_x U^n + \Delta t^2 \tilde{A_{xx}} U^n + O\left(\Delta t^3\right) \\ W &= \left(1 - 2\Delta_t \tilde{A}_y\right)^{-1} U^n = U^n + 2\Delta t \tilde{A}_y U^n + \Delta t^2 \tilde{A_{yy}} U^n + O\left(\Delta t^3\right) \end{aligned} \tag{40}$$

Introduce Eqs. (18 and 38) into (40):

$$\begin{aligned} U^{n+1} &= U^n + \Delta_t \left(A_x + A_y\right) U^n + \Delta_t \left(L_x\, U^n\, \frac{\Delta_x^2}{24} + L_y\, U^n\, \frac{\Delta_y^2}{24} \right) \\ &+ \frac{\Delta_t^2}{2} \left(A_{xx} + A_{yy}\right) U^n + O\left(\Delta t^3\right) \end{aligned} \tag{41}$$

According to the left side of Eq. (12), the numerical approximation of the scale (time) derivative

$$\dot{\tilde{U}}^{n+1/2} = \frac{U^{n+1} - U^n}{\Delta t} + O\left(\Delta t^2\right) \tag{42}$$

Introduce Eq. (41) into Eq. (42) and assume equal qrid in x and y:

$$\Delta x = \Delta y = h \tag{43}$$

$$\dot{\tilde{U}}^{n+1/2} = \left(A_x + A_y\right) U^n + \frac{h^2}{24} \left(L_x + L_y\right) U^n + \frac{\Delta_t}{2} \left(A_{xx} + A_{yy}\right) U^n \tag{44}$$

Expand $\dot{\tilde{U}}^{n+1/2}$ into the Taylor series in the proximity of scale level n:

$$\dot{\tilde{U}}^{n+1/2} = \dot{\tilde{U}}^n + \frac{\Delta t}{2} \ddot{U}^n + O\left(\Delta t^2\right) \tag{45}$$

and introduce the result into Eq. (44):

$$\dot{\tilde{U}}^n + \frac{\Delta t}{2} \ddot{U}^n = \left(A_x + A_y\right) U^n + \frac{h^2}{24} \left(L_x + L_y\right) U^n + \frac{\Delta_t}{2} \left(A_{xx} + A_{yy}\right) U^n \tag{46}$$

Assume that the original Beltrami PDE holds. Then it follows from Eq. (8):

$$\ddot{U} = \left(A_x + A_y\right)^2 U = \left(A_{xx} + A_{xy} + A_{yx}\, A_{yy}\right) U \tag{47}$$

Note that A_{xy} and A_{yx} are not identical:

$$\begin{aligned} A_{xy} &= \frac{\partial}{\partial y}\left(\frac{h}{2}\frac{\partial A_x}{\partial y}\right) + \frac{h}{2}\frac{\partial^2 A_x}{\partial y^2} \\ A_{yx} &= \frac{\partial}{\partial x}\left(\frac{h}{2}\frac{\partial A_y}{\partial x}\right) + \frac{h}{2}\frac{\partial^2 A_y}{\partial x^2} \end{aligned} \tag{48}$$

where A_x and A_y are differential operators defined in Eq. (9). Introduce Eqs. (8 and 47) into (46):

$$\tilde{\dot{U}} - \dot{U} = \frac{h^2}{24}\left(L_x + L_y\right) U + \frac{\Delta_t}{2}\left(A_{xy} + A_{yx}\right) U \tag{49}$$

where $\dot{U}$ is the scale derivative of the pixel function, $\tilde{\dot{U}}$ is its numerical approximation, and the local truncation error E is the discrepancy:

$$E = \tilde{\dot{U}} - \dot{U} = O\left(\Delta t + s^2\right) \quad \text{where} \quad s = \Delta x = \Delta y \tag{50}$$

Summarizing the above derivation, we outline four sources for the local truncation error of the implicit difference scheme:

1. Finite difference approximationin of space derivatives
2. Replacement of half-integer values of the edge indicator function with integer values
3. Backward (non-central) differences in time
4. High order terms of additive operator split

7 Simulation Results for Gray Level Images

A series of numerical simulations was run to study the accuracy of the implicit scheme for the Beltrami smoothing flow. For this, the acceleration factor f was introduced. Factor f shows the ratio of the acual scale step in the implicit scheme to the ultimate value of the scale step for the explicit scheme. Recall that for the square grid $\Delta x = \Delta y = s$, the ultimate value $\Delta t_{\max} = s^2/4$. Assuming further that the unit of length is defined as the

distance between the pixels, $s = 1$, and $\Delta t_{\max} = 0.25$. Simulations were run for different values of f, starting from $f = 1$ and up to $f = 200$. The results are presented in Fig. 2. In each case, the computational time was measured. As wee see, the implicit scheme is always stable, but for $f \gg 20$, the accuracy of processing may be insufficient. Perhaps, the average recommended values are $f = 10 \ldots 20$. Note that for a single step, the implicit scheme takes approximately twice more computational time than the explicit scheme.

Next series of numerical simulations was carried out to study the edge enhancemeent effect for gray level images. In all cases of the secon series, the acceleration factor was 10. A normalized reaction-diffusion smoothing governed by PDE (51) was applied to a medical image:

$$\frac{\partial U}{\partial t} = \cos\beta \ \nabla h \cdot \nabla U + \sin\beta \ h \ \nabla^2 U \tag{51}$$

The first term on the right side of Eq. (51) is a reaction term, while the second is a diffusion term. β is a parameter presenting the relative contribution of reaction and diffusion. The reaction term is responsible for edge enhancement, while the diffusion term - for smoothing the random noise. β varies from 0 to 90^o. Results of edge enhancement for different values of β are presented in Fig. 3. $\beta = 0$ corrsponds to a pure reaction, $\beta = 45^o$ is a nonlinear diffusion flow (until the constant scale factor), $\beta = \arctan 2 \approx 63.4^o$ is the Beltrami flow, and $\beta = 90^o$ is a normalized 'linear' diffusion (ceases to be linear after normalization).

According to Eq. (51), the edge enhancement effect should decay as β increases. Indeed, we see that the edge enhancement is stronger for the nonlinear diffusion flow ($\beta = 45^o$) than for the Beltrami flow ($\beta = 63.4^o$).

8 Beltrami Smoothing for Color Images

The Beltrami flow for color images is governed by the following set of PDE [4, 5]:

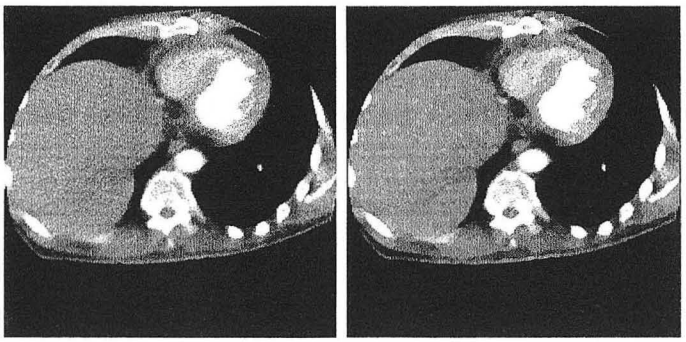
$$\frac{\partial I_i}{\partial t} = \frac{\dfrac{\partial P_i}{\partial x} + \dfrac{\partial Q_i}{\partial y}}{g} - \frac{\dfrac{\partial g}{\partial x} P_i + \dfrac{\partial g}{\partial y} Q_i}{2\, g^2} \tag{52}$$

(a) Initial Image (b) $f = 1,\ t = 107$ s (c) $f = 2,\ t = 53.9$ s

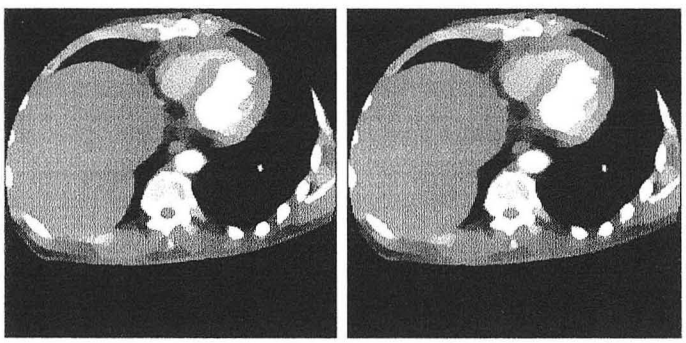
(d) $f = 5,\ t = 21.5$ s (e) $f = 10,\ t = 10.7$ s (f) $f = 20,\ t = 5.38$ s

(g) $f = 50,\ t = 2.14$ s (h) $f = 100,\ t = 1.07$ s (i) $f = 200,\ t = 0.54$ s

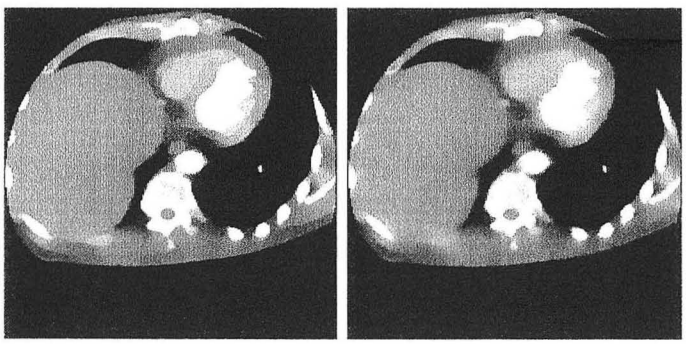
Figure 2: Beltrami Smoothing, Implicit Difference Scheme, Scale 250



(a) Initial Image (b) $\beta = 0$



(c) $\beta = 30^o$ (d) $\beta = 45^o$



(e) $\beta \approx 63.4^o$ (f) $\beta = 30^o$

Figure 3: Edge Enhancement of Medical Image, Scale 250

where $i = 1, 2, 3$ is the number of color (red, green, blue), I_i is the corresponding pixel value and P, Q are defined by:

$$P_i = g_{22} \frac{\partial I_i}{\partial x} - g_{12} \frac{\partial I_i}{\partial y} \qquad Q_i = -g_{12} \frac{\partial I_i}{\partial x} + g_{11} \frac{\partial I_i}{\partial y} \tag{53}$$

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:

$$\boldsymbol{G} = \begin{bmatrix} g_{11} & g_{12} \\ g_{12} & g_{22} \end{bmatrix} \tag{54}$$

$$\begin{aligned} g_{11} &= 1 + \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial x} \right)^2 \qquad & g_{22} &= 1 + \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial y} \right)^2 \\ g_{12} &= \sum_{j=1}^{3} \frac{\partial I_j}{\partial x} \frac{\partial I_j}{\partial y} \qquad & g &= \det \boldsymbol{G} = g_{11}\, g_{22} - g_{12}^2 \end{aligned} \tag{55}$$

9 Nonlinear Diffusion Flow for Color Images

So far, instead of Beltrami flow (52), consider a simplified smoothing flow, which is an analog for the nonlinear diffusion for gray level image:

$$\frac{\partial I_i}{\partial t} = \frac{\dfrac{\partial P_i}{\partial x} + \dfrac{\partial Q_i}{\partial y}}{g} - \frac{\dfrac{\partial g}{\partial x} P_i + \dfrac{\partial g}{\partial y} Q_i}{g^2} \tag{56}$$

Note that in Eq. (56), there is no factor 2 in the denominator of the last term. Eq. (56) becomes easier to operate in this analysis. It may be rearranged as:

$$\frac{\partial I}{\partial t} = \frac{\partial}{\partial x} \left(\frac{P}{g} \right) + \frac{\partial}{\partial y} \left(\frac{Q}{g} \right) \tag{57}$$

Define vector $\boldsymbol{S}_i$ of length 2, whose components are P_i/g and Q_i/g: Then, according to the above,

$$\boldsymbol{S}_i = \boldsymbol{G}^{-1} \nabla I_i = \boldsymbol{R} \nabla I_i \tag{58}$$

where $\boldsymbol{R}$ is a notation for an inverse of $\boldsymbol{G}$. The flow becomes:

$$\frac{\partial I_i}{\partial t} = \nabla \cdot \boldsymbol{S}_i = \nabla \cdot (\boldsymbol{R} \nabla I_i) \tag{59}$$

Note that there are three pixel value gradient vectors ∇I_i (for each color component), three vectors $\boldsymbol{S}_i$. 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.

Opening the brackets in the last equation, we obtain a reaction term (includes the first dericatives of pixel values), and a diffusion term (includes the second derivatives).

$$\frac{\partial I_i}{\partial t} = (\nabla \cdot \boldsymbol{R}) \cdot \nabla I_i + \boldsymbol{R} \cdot \nabla \nabla I_i \tag{60}$$

Note that 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 a matrix of second derivatives:

$$\nabla \nabla I_i = \begin{bmatrix} I^i_{xx} & I^i_{xy} \\ I^i_{xy} & I^i_{yy} \end{bmatrix} \tag{61}$$

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

10 Eigenvalues of Edge Indicator Matrix

Considerv the nonlinear diffusion flow

$$\frac{\partial I_i}{\partial t} = \nabla \cdot (\boldsymbol{R} \nabla I_i) \tag{62}$$

One may get an analogy with the scalar gray level case if the action of matrix $\boldsymbol{R}$ on vector $nablaI_i$ is replaced by the action of the scalar number on that vector. This number is an eigenvalue.

For this, we de-compose the dradient vector ∇I_i into the basis of principal directions $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$.

$$\nabla I_i \;=\; k_1 \boldsymbol{V}_1 \;+\; k_2 \boldsymbol{V}_2 \tag{63}$$

where $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ are normalized eigenvectors (say, of unit length) of the Edge Indicator Tensor $\boldsymbol{R}$. Assume λ_1 and λ_2 are eigenvalues, corresponding to these eigenvectors:

$$\boldsymbol{R}\boldsymbol{V}_k \;=\; \lambda_k \boldsymbol{V}_k \qquad\qquad k \;=\; 1,\; 2 \tag{64}$$

Consider the expression inside the brackets on the right side of Eq. (62).

$$\boldsymbol{R}\nabla I_i \;=\; \boldsymbol{R}\,(k_1 \boldsymbol{V}_1 \;+\; k_2 \boldsymbol{V}_2) \;=\; k_1 \boldsymbol{R}\boldsymbol{V}_1 \;+\; k_2 \boldsymbol{R}\boldsymbol{V}_2 \;=\; k_1 \lambda_1 \boldsymbol{V}_1 \;+\; k_2 \lambda_2 \boldsymbol{V}_2 \tag{65}$$

Note that both λ_k are real positive numbers since matrix $\boldsymbol{R}$ is symmetric and positive definite. Now, let us find these eigenvalues:

$$\det \begin{bmatrix} \dfrac{1 \;+\; \sum_{j=1}^{3} \left(\dfrac{\partial I_j}{\partial y}\right)^2}{g} \;-\; \lambda_k & \dfrac{-\sum_{j=1}^{3} \dfrac{\partial I_j}{\partial x}\,\dfrac{\partial I_j}{\partial y}}{g} \\ \dfrac{-\sum_{j=1}^{3} \dfrac{\partial I_j}{\partial x}\,\dfrac{\partial I_j}{\partial y}}{g} & \dfrac{1 \;+\; \sum_{j=1}^{3} \left(\dfrac{\partial I_j}{\partial x}\right)^2}{g} \;-\; \lambda_k \end{bmatrix} \;=\; 0 \tag{66}$$

This leads to a quadratic equation:

$$\left[1 \;+\; \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial x}\right)^2 \;-\; g\lambda_k\right] \cdot \left[1 \;+\; \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial y}\right)^2 \;-\; g\lambda_k\right] \;-\; \left(\sum_{j=1}^{3} \frac{\partial I_j}{\partial x}\,\frac{\partial I_j}{\partial y}\right)^2 \;=\; 0 \tag{67}$$

Introduce a new notation:

$$\Lambda_k \;=\; g\,\lambda_k \qquad\qquad k \;=\; 1,\; 2 \tag{68}$$

and recall that

$$g = \det \boldsymbol{G} = 1 + \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial x} \right)^2 + \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial y} \right)^2 + \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial x} \right)^2 \cdot \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial y} \right)^2 - \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial x} \frac{\partial I_j}{\partial y} \right)^2 \tag{69}$$

Eq. (67) leds to:

$$\Lambda_k^2 + \left[2 + \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial x} \right)^2 + \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial y} \right)^2 \right] \Lambda_k + g = 0 \tag{70}$$

The roots of Eq.(69) are:

$$\Lambda_{1,2} = 1 + \frac{\sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial x} \right)^2 + \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial y} \right)^2}{2} \pm \sqrt{\frac{\left[\sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial x} \right)^2 - \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial y} \right)^2 \right]^2}{4} + \left(\sum_{j=0}^{3} \frac{\partial I_j}{\partial x} \frac{\partial I_j}{\partial y} \right)^2} \tag{71}$$

Eq. (71) may be alternatively presented in the form:

$$\Lambda_{1,2} = 1 + \frac{\sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial x} \right)^2 + \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial y} \right)^2}{2} \pm \sqrt{\frac{\left[\sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial x} \right)^2 + \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial y} \right)^2 \right]^2}{4} - A} \tag{72}$$

where

$$A = \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial x} \right)^2 \cdot \sum_{j=1}^{3} \left(\frac{\partial I_j}{\partial y} \right)^2 - \left(\sum_{j=1}^{3} \frac{\partial I_j}{\partial x} \frac{\partial I_j}{\partial y} \right)^2 \tag{73}$$

Expanding powers and products, Eq. (73) may be rearranged in the following form:

$$
\begin{aligned}
A \;=\; & \left[\left(\frac{\partial I_1}{\partial x}\right)^2 + \left(\frac{\partial I_2}{\partial x}\right)^2 + \left(\frac{\partial I_3}{\partial x}\right)^2\right] \cdot \left[\left(\frac{\partial I_1}{\partial y}\right)^2 + \left(\frac{\partial I_2}{\partial y}\right)^2 + \left(\frac{\partial I_3}{\partial y}\right)^2\right] - \\
& \left(\frac{\partial I_1}{\partial x}\frac{\partial I_1}{\partial y} + \frac{\partial I_2}{\partial x}\frac{\partial I_2}{\partial y} + \frac{\partial I_3}{\partial x}\frac{\partial I_3}{\partial y}\right)^2 = \\
& \left(\frac{\partial I_1}{\partial x}\right)^2\left(\frac{\partial I_1}{\partial y}\right)^2 + \left(\frac{\partial I_2}{\partial x}\right)^2\left(\frac{\partial I_1}{\partial y}\right)^2 + \left(\frac{\partial I_3}{\partial x}\right)^2\left(\frac{\partial I_1}{\partial y}\right)^2 + \\
& \left(\frac{\partial I_1}{\partial x}\right)^2\left(\frac{\partial I_2}{\partial y}\right)^2 + \left(\frac{\partial I_2}{\partial x}\right)^2\left(\frac{\partial I_2}{\partial y}\right)^2 + \left(\frac{\partial I_3}{\partial x}\right)^2\left(\frac{\partial I_2}{\partial y}\right)^2 + \\
& \left(\frac{\partial I_1}{\partial x}\right)^2\left(\frac{\partial I_3}{\partial y}\right)^2 + \left(\frac{\partial I_2}{\partial x}\right)^2\left(\frac{\partial I_3}{\partial y}\right)^2 + \left(\frac{\partial I_3}{\partial x}\right)^2\left(\frac{\partial I_3}{\partial y}\right)^2 - \\
& \left(\frac{\partial I_1}{\partial x}\frac{\partial I_1}{\partial y}\right)^2 - \left(\frac{\partial I_2}{\partial x}\frac{\partial I_2}{\partial y}\right)^2 - \left(\frac{\partial I_3}{\partial x}\frac{\partial I_3}{\partial y}\right)^2 - \\
& 2\,\frac{\partial I_1}{\partial x}\frac{\partial I_1}{\partial y}\frac{\partial I_2}{\partial x}\frac{\partial I_2}{\partial y} - 2\,\frac{\partial I_2}{\partial x}\frac{\partial I_2}{\partial y}\frac{\partial I_3}{\partial x}\frac{\partial I_3}{\partial y} - 2\,\frac{\partial I_3}{\partial x}\frac{\partial I_3}{\partial y}\frac{\partial I_1}{\partial x}\frac{\partial I_1}{\partial y} = \\
& \left(\frac{\partial I_1}{\partial x}\frac{\partial I_2}{\partial y} - \frac{\partial I_2}{\partial x}\frac{\partial I_1}{\partial y}\right)^2 + \left(\frac{\partial I_2}{\partial x}\frac{\partial I_3}{\partial y} - \frac{\partial I_3}{\partial x}\frac{\partial I_2}{\partial y}\right)^2 + \\
& \left(\frac{\partial I_3}{\partial x}\frac{\partial I_1}{\partial y} - \frac{\partial I_1}{\partial x}\frac{\partial I_3}{\partial y}\right)^2
\end{aligned}
\tag{74}
$$

Thus, A presents a sum of squares of lengts for cross-products of pirs of gradients $\nabla I_m \times \nabla I_{m+1}$.

$$
A \;=\; (\nabla I_1 \times \nabla I_2)^2 \;+\; (\nabla I_2 \times \nabla I_3)^2 \;+\; (\nabla I_3 \times \nabla I_1)^2 \tag{75}
$$

11 Basic Assumption in Proximity of Edge

Let us define the edge. For the gray level image, the edge is a line which divides the plane into segments with different pixel value. In other words,

when passing across the edge, the pixel value changes in a discontinuour (or continuous, but very rapid) manner. The magnitude of the gradient vector is large (although not necessarily infinite), 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 gradient. The direction of this gradient is normal to the egde. The directions of other color gradients are collinear to this direction: either the same, or opposite, provided the magnitudes of gradients of these other components are also large. For the component(s) of small-magnitude gradient, the direction of gradient does not matter. Thus, in the close proximity of the edge, the value A in Eq. (75) will be small because all cross-products of gradients vanish or almost vanish: either since the components of the cross-product are collinear, or due to the fact that one of them or both have small magnitude.

12 Eigenvalues and Eigenvectors of Edge Indicator Matrix in Proximity of Edge

Since the cross-products are small in the proximity of the egde, the value A in Eq. (72) may be neglected. The eigenvalues become:

$$\lambda_1 \approx \frac{1}{g} \qquad \lambda_2 \approx 1 \qquad g \approx 1 + \sum_{j=0}^{3} \left(\frac{\partial I_j}{\partial x} \right)^2 + \sum_{j=0}^{3} \left(\frac{\partial I_j}{\partial y} \right)^2 \tag{76}$$

Find the principal direction of the edge indicator matrix $\boldsymbol{R}$, corrsponding to the smaller eigenvalue $1/g$:

$$\begin{bmatrix} \dfrac{1 + \sum_{j=1}^{3}\left(\dfrac{\partial I_j}{\partial y}\right)^2}{g} - \dfrac{1}{g} & \dfrac{-\sum_{j=1}^{3}\dfrac{\partial I_j}{\partial x}\dfrac{\partial I_j}{\partial y}}{g} \\ \dfrac{-\sum_{j=1}^{3}\dfrac{\partial I_j}{\partial x}\dfrac{\partial I_j}{\partial y}}{g} & \dfrac{1 + \sum_{j=1}^{3}\left(\dfrac{\partial I_j}{\partial x}\right)^2}{g} - \dfrac{1}{g} \end{bmatrix} \begin{bmatrix} V_1^x \\ V_1^y \end{bmatrix} = 0 \tag{77}$$

It follows form Set (77)

$$\frac{V_1^y}{V_1^x} = \frac{\sum_{j=0}^{3}\left(\dfrac{\partial I_j}{\partial y}\right)^2}{\sum_{j=1}^{3}\dfrac{\partial I_j}{\partial x}\dfrac{\partial I_j}{\partial y}} \qquad \frac{V_1^y}{V_1^x} = \frac{\sum_{j=1}^{3}\dfrac{\partial I_j}{\partial x}\dfrac{\partial I_j}{\partial y}}{\sum_{j=1}^{3}\left(\dfrac{\partial I_j}{\partial x}\right)^2} \tag{78}$$

Each equation of set (78) follows from the correponding equation of set (77). However, the two equations in set (77) are linearly dependent, so that both results for V_1^y/V_1^x should be the same. Recall that

$$A \approx 0 \quad \longrightarrow \quad \sum_{j=1}^{3}\frac{\partial I_j}{\partial x}\frac{\partial I_j}{\partial y} \approx \sqrt{\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial x}\right)^2 \cdot \sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial y}\right)^2} \tag{79}$$

Introducing Eq. (79) into any equation of Set (78), we get the same result:

$$\sqrt{\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial y}\right)^2 \Big/ \sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial x}\right)^2} \tag{80}$$

Since we assumed that all gradients (of essential magnitude) have approximately collinear directions in the proximity of the edge, it follows from Eq. (80) that the principal direction $\boldsymbol{V}_1$ coincides with these gradients and is normal to the edge. Now let us find the second principal direction, corresponding

to a larger eigenvalue 1.

$$\begin{bmatrix} \dfrac{1 + \sum_{j=1}^{3}\left(\dfrac{\partial I_j}{\partial y}\right)^2}{g} - 1 & \dfrac{-\sum_{j=1}^{3}\dfrac{\partial I_j}{\partial x}\dfrac{\partial I_j}{\partial y}}{g} \\ \dfrac{-\sum_{j=1}^{3}\dfrac{\partial I_j}{\partial x}\dfrac{\partial I_j}{\partial y}}{g} & \dfrac{1 + \sum_{j=1}^{3}\left(\dfrac{\partial I_j}{\partial x}\right)^2}{g} - 1 \end{bmatrix} \begin{bmatrix} V_2^x \\ V_2^y \end{bmatrix} = 0 \tag{81}$$

It follows form Set (81)

$$\frac{V_1^x}{V_1^y} = -\frac{\sum_{j=0}^{3}\left(\dfrac{\partial I_j}{\partial x}\right)^2}{\sum_{j=1}^{3}\dfrac{\partial I_j}{\partial x}\dfrac{\partial I_j}{\partial y}} = -\frac{\sum_{j=1}^{3}\dfrac{\partial I_j}{\partial x}\dfrac{\partial I_j}{\partial y}}{\sum_{j=1}^{3}\left(\dfrac{\partial I_j}{\partial y}\right)^2} = -\sqrt{\sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial x}\right)^2 \Big/ \sum_{j=1}^{3}\left(\frac{\partial I_j}{\partial y}\right)^2} \tag{82}$$

In the proximity of the edge, all three forms of Eq. (82) are "approximately equivalent". The second eigenvector is dierected along the edge. As expected, it is normal to the first eigenvector:

$$\boldsymbol{V}_1 \cdot \boldsymbol{V}_2 = 0 \tag{83}$$

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 $\lambda_2 \approx 1$. Let us define the egde direction **exactly** as an eigenvector of the indicator matrix.

13 Governing PDE in the Poximity of Edge

Consider Eq. (65). Since the edge is normal to the gradient (provided the gradient magnitude is not negligibly small),

$$k_1 \approx |\nabla I_i| \qquad k_2 \approx 0 \tag{84}$$

$$\boldsymbol{R}\nabla I_i \;=\; k_1\lambda_1\boldsymbol{V}_1 \;+\; k_2\lambda_2\boldsymbol{V}_2 \;\approx\; \frac{\nabla I_i}{g} \tag{85}$$

where g is given by Eq. (76). In the proximity of the edge, the governig PDE for smoothing the color image comes to:

$$\frac{\partial I_i}{\partial t} \;=\; \nabla\cdot(\boldsymbol{R}\nabla I_i) \;=\; \nabla\cdot\left(\frac{\nabla I_i}{g}\right) \;=\; \nabla\left(\frac{1}{g}\right)\cdot\nabla I_i \;+\; \frac{\nabla^2 I_i}{g} \tag{86}$$

Now, recall that Eq. (86) describes the nonlinear diffusion flow which differs from the Beltrami flow. It follows from Eqs. (52, refE5 and 86) that the Beltrami flow has an additional reaction term:

$$\frac{\partial I_i}{\partial t} \;=\; \nabla\left(\frac{1}{g}\right)\cdot\nabla I_i \;+\; \frac{\nabla^2 I_i}{g} \;+\; \frac{\dfrac{\partial g}{\partial x}P_i + \dfrac{\partial g}{\partial y}Q_i}{2\,g^2} \tag{87}$$

Re-arrange the last term in Eq. (87) to a tensor form:

$$\begin{aligned}\frac{\dfrac{\partial g}{\partial x}P_i + \dfrac{\partial g}{\partial y}Q_i}{2\,g^2} \;&=\; -\,\frac{1}{2}\left[\frac{\partial}{\partial x}\left(\frac{1}{g}\right)P_i \;+\; \frac{\partial}{\partial y}\left(\frac{1}{g}\right)Q_i\right] \\ &=\; -\,\frac{1}{2}\,\nabla\left(\frac{1}{g}\right)\cdot g\,\boldsymbol{R}\,\nabla I_i\end{aligned} \tag{88}$$

Now apply Eq. (85) for the proximity of the edge:

$$g\,\boldsymbol{R}\,\nabla I_i \;\approx\; \nabla I_i \tag{89}$$

The governing equation (87) for Beltrami flow comes to:

$$\frac{\partial I_i}{\partial t} \;\approx\; \frac{1}{2}\,\nabla\left(\frac{1}{g}\right)\cdot\nabla I_i \;+\; \frac{\nabla^2 I_i}{g} \;=\; \frac{1}{2}\,\nabla\cdot\left(\frac{\nabla I_i}{g}\right) \;+\; \frac{1}{2}\,\frac{\nabla^2 I_i}{g} \tag{90}$$

However, this coincides exactly with the governing PDE for the Beltrami smoothing of 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. This means that the gradient of $1/g$ is directed outside the thin pass of the edge. The gradient of the pixel value is also normal

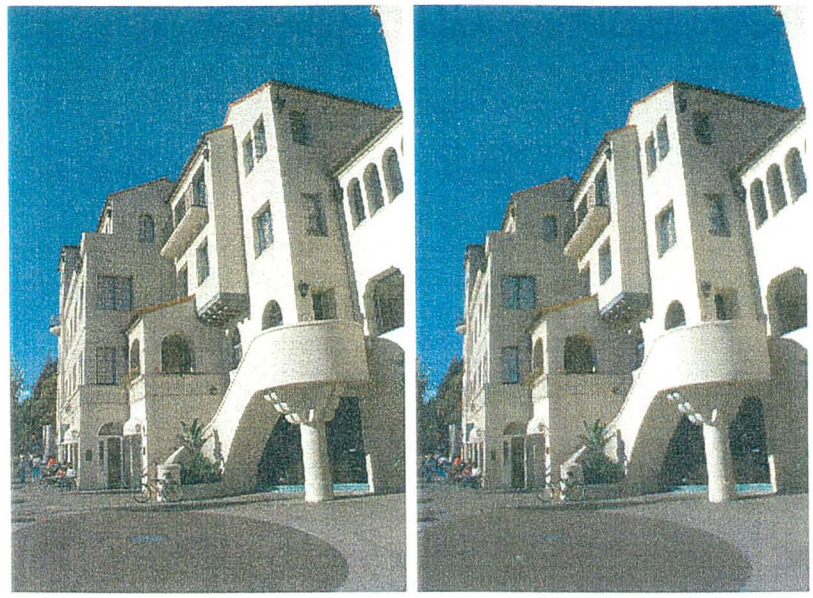
to the edge, but its direction coincides with the gradient of $1/g$ for larger pixel values, and is opposite to that direction for smaller pixelvalues. The reactive component of smoothing becomes positive for larger pixel values and negative - for smaller pixel in the proximity of the edge. The larger values become even larger, and the samll values become even smaller. The edge enhances, it becomes more sharp.

Note that the simplified decoupled form of the Beltrami smoothing may be recommended for numerical computations. Close to the edge this form is justified because the coupling between x and y components of gradient becomes weak. Far from the edge, the decoupled form brings a definite inaccuracy, and the smoothing is no longer exactly Belltrami smoothing. 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 PDE (90) 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 even much better saving of the computational time.

14 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.

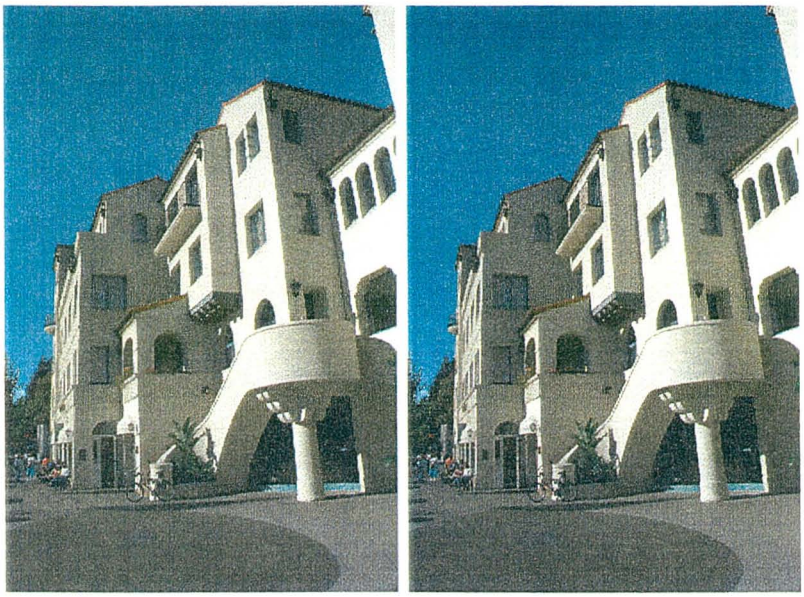
Fig. 4 presents (a) the initial image and the smoothing simulation results for the Beltrami filtering of color image using (b) coupled explicit difference scheme with a full edge indicator matrix, (c) explicit difference scheme with decoupled governing equation and the eigenvalue instead of the mathix, and (d) implicit difference scheme with decoupled governing equation and acceleration factor $f = 1$. Fig. 5 prsents the simulation results for implicit scheme with different values of acceleration factor $f = 2,\ 5,\ 10$ and 20.



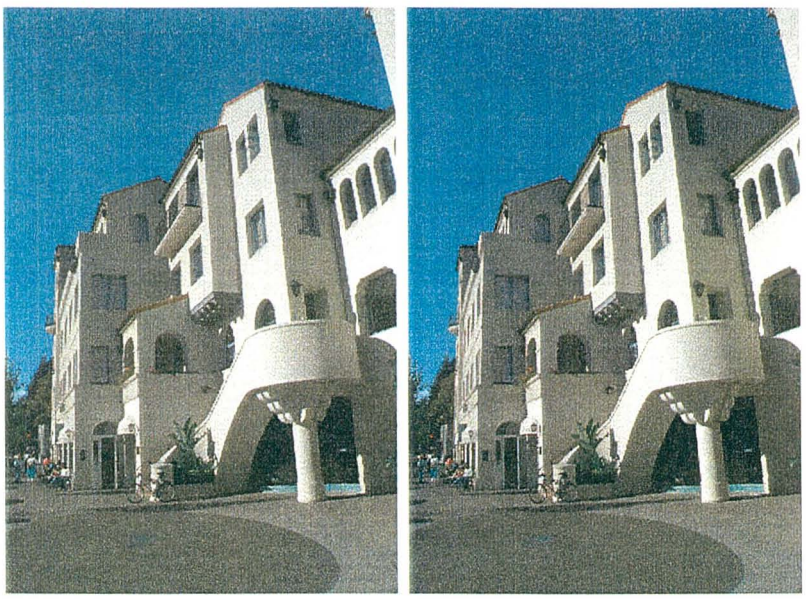
(a) Initial Image (b) Explicit Coupled Scheme

(c) Expl. Decoupled Scheme (d) Implicit Scheme, $f = 1$

Figure 4: Beltrami Smoothing of Color Images, Scale 100



(a) Implicit Scheme, $f = 2$ (b) Implicit Scheme, $f = 5$



(c) Implicit Scheme, $f = 10$ (d) Implicit Scheme, $f = 20$

Figure 5: Fast Beltrami Smoothing of Color Images, Scale 100

15 Closing Remarks

1. 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.

2. Basing on this approach, the uncondiotionally stable difference scheme is developed

3. 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.

4. 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.

5. The approach may be applied also for mean curvature flow by a similar way, with a different edge indicator function.

6. 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 to replace the action of the edge indicator matrix on the gradient vector by the action of its eigenvalue on that vector.

7. The main assumption of the above derivation is that the cross-products of the dradients 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.

8. 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] 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).

[2] 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.

[3] 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.

[4] 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.

[5] N. Sochen, R. Kimmel and R. Malladi. "A genearal framework for low level vision". *IEEE Transactions*, **Vol. 7, No. 3**, 1998.

[6] J. Weickert. *Anisotropic Diffusion in Image Processing.* Ph. D. Thesis, Kaiserslautern University, 1996.

Contents

List of Figures

Ernest Orlando Lawrence Berkeley National Laboratory
One Cyclotron Road | Berkeley, California 94720

Prepared for the U.S. Department of Energy under Contract No. DE-AC03-76SF00098