Efficient Beltrami Filtering of Color Images via Vector Extrapolation

Lorina Dascal, Guy Rosman, and Ron Kimmel

Computer Science Department, Technion,
Institute of Technology, Haifa 32000, Israel

Abstract. The Beltrami image flow is an effective non-linear filter, often used in color image processing. It was shown to be closely related to the median, total variation, and bilateral filters. It treats the image as a 2D manifold embedded in a hybrid spatial-feature space. Minimization of the image area surface yields the Beltrami flow. The corresponding diffusion operator is anisotropic and strongly couples the spectral components. Thus, there is so far no implicit nor operator splitting based numerical scheme for the PDE that describes Beltrami flow in color. Usually, this flow is implemented by explicit schemes, which are stable only for very small time steps and therefore require many iterations. At the other end, vector extrapolation techniques accelerate the convergence of vector sequences, without explicit knowledge of the sequence generator. In this paper, we propose to use the minimum polynomial extrapolation (MPE) and reduced rank extrapolation (RRE) vector extrapolation methods for accelerating the convergence of the explicit schemes for the Beltrami flow. Experiments demonstrate their stability and efficiency compared to explicit schemes.

Keywords: *Color processing, geometric heat flow, Laplace-Beltrami operator*

1 Introduction

Nonlinear partial differential diffusion equations are used extensively for various image processing applications. The Beltrami framework was first introduced in [20], then in [21], followed by [26]. This filter was applied for edge preserving denoising and deblurring for signals and especially multi-channel images, see for example [1]. In this paper we demonstrate an efficient scheme for computing the Beltrami color flow. This flow is usually implemented by a discrete approximation of a partial differential equation (PDE). Standard explicit finite difference schemes for heat equations require small time-steps for stability, that lead to a large number of iterations required for convergence to the desired solution.

The additive operator splitting (AOS) scheme was first developed for solving the Navier-Stokes equations [10, 11]. It was later used in [24] for implementing the regularized Perona-Malik filter [4]. The AOS scheme is first order in time, semi-implicit, and unconditionally stable with respect to its time step. It can be used for single channel diffusion filters, like the Perona-Malik [12], the Total variation (TV) [14], or the anisotropic diffusion filter for grey-level images [25]. Unfortunately, due to the strong coupling and its anisotropic nature, splitting is difficult for the Beltrami color operator. In fact, so far, there is no PDE based implicit scheme for Beltrami flow in color.

In [22] the short time kernel for the Beltrami operator was computed in order to approximate the Beltrami flow. This method is still computationally demanding, as computing the kernel operation involves geodesic distance computation around each pixel. The bilateral operator was studied in different contexts (see for example [18], [23], [19], [7], [2]), and can be shown to be an approximation of the Beltrami kernel.

In this paper we propose to apply vector extrapolation methods to accelerate the convergence rate of standard explicit schemes for the Beltrami flow in color. The minimum polynomial extrapolation algorithm (MPE) of Cabay and Jackson [3] and the reduced rank extrapolation algorithm (RRE) of Eddy [6] are two vector extrapolation methods derived in order to accelerate the convergence of vector sequences. They are obtained from an iterative solution of linear and nonlinear systems of equations. Both MPE and RRE algorithms are detailed in Section 4 with their respective definitions as solutions for least squares problems. An efficient solution is achieved by the modified Gram-Schmidt algorithm [16]. These vector extrapolation methods can be computed directly from the elements of the sequence. Furthermore, unlike alternative acceleration techniques, they can be applied not only to linearly generated sequences, but also to nonlinear ones. This allows us to apply the RRE/MPE methods for accelerating the convergence of the vector sequence generated in the explicit scheme for the Beltrami geometric flow.

We demonstrate the efficiency and accuracy of the vector extrapolation methods in color image processing applications such as: scale-space analysis, denoising, and deblurring.

This paper is organized as follows: Section 2 gives a brief summary of the Beltrami framework. In Section 3 we review approximations based on standard explicit finite difference schemes. Section 4 describes the minimal polynomial extrapolation (MPE) and the reduced rank extrapolation (RRE) algorithms, the two methods we use for accelerating the convergence of the standard explicit scheme. In Section 5 we apply the RRE algorithm to our Beltrami color flow and demonstrate the resulting speed-up. Section 6 concludes the paper.

2 The Beltrami Framework

Let us briefly review the Beltrami framework for non-linear diffusion in computer vision [9, 20, 21, 26]. We represent images as embedding maps of a Riemannian manifold in a higher dimensional space. We denote the map by $U : \Sigma \rightarrow M$, where Σ is a two-dimensional surface, with (σ^1, σ^2) denoting coordinates on it. M is the spatial-feature manifold, embedded in $\mathbb{R}^{d+2}$, where d is the number of image channels. For example, a gray-level image can be represented as a 2D surface embedded in $\mathbb{R}^3$. The map U in this case is $U(\sigma^1, \sigma^2) = (\sigma^1, \sigma^2, I(\sigma^1, \sigma^2))$, where I is the image intensity. For color images, U is given by $U(\sigma^1, \sigma^2) = (\sigma^1, \sigma^2, I^1(\sigma^1, \sigma^2), I^2(\sigma^1, \sigma^2), I^3(\sigma^1, \sigma^2))$, where I^1, I^2, I^3 are the three components of the color vector.

Next, we choose a Riemannian metric on this surface. The canonical choice of coordinates in image processing is Cartesian (we denote them here by x^1 and x^2). For such a choice, which we follow in the rest of the paper, we identify $\sigma^1 = x^1$ and $\sigma^2 = x^2$. In this case, σ^1 and σ^2 are the image coordinates. We denote the elements of the inverse of the metric by superscripts g^{ij}, and the determinant by $g = \det(g_{ij})$. Once images are defined as embedding of Riemannian manifolds, it is natural to look for a measure on this space of embedding maps.

Denote by (Σ, g) the image manifold and its metric, and by (M, h) the space-feature manifold and its metric. Then, the functional $S[U]$ attaches a real number to a map $U : \Sigma \to M$,

$$S[U, g_{ij}, h_{ab}] = \int d^m\sigma \sqrt{g} ||dU||^2_{g,h}, \tag{1}$$

where m is the dimension of Σ, g is the determinant of the image metric, and the range of indices is $i, j = 1, 2, \ldots \dim(\Sigma)$ and $a, b = 1, 2, \ldots \dim(M)$. The integrand $||dU||^2_{g,h}$ is expressed in a local coordinate system by $||dU||^2_{g,h} = (\partial_{x_i} U^a) g^{ij} (\partial_{x_j} U^b) h_{ab}$. This functional, for $\dim(\Sigma) = 2$ and $h_{ab} = \delta_{ab}$, was first proposed by Polyakov [13] in the context of high energy physics, in the theory known as *string theory.* The elements of the induced metric for color images with Cartesian color coordinates are

$$G = (g_{ij}) = \begin{pmatrix} 1 + \beta^2 \sum_{a=1}^3 (U^a_{x_1})^2 & \beta^2 \sum_{a=1}^3 U^a_{x_1} U^a_{x_2} \\ \beta^2 \sum_{a=1}^3 U^a_{x_1} U^a_{x_2} & 1 + \beta^2 \sum_{a=1}^3 (U^a_{x_2})^2 \end{pmatrix},$$

where a subscript of U denotes partial derivation and the parameter $\beta > 0$ determines the ratio between the spatial and spectral (color) distances. Using standard methods in the calculus of variations, the Euler-Lagrange equations with respect to the embedding (assuming Euclidean embedding space) are

$$0 = -\frac{1}{\sqrt{g}} h^{ab} \frac{\delta S}{\delta U^b} = \underbrace{\frac{1}{\sqrt{g}} \operatorname{div}(D \nabla U^a)}_{\Delta_g U^a}, \tag{2}$$

where the matrix $D = \sqrt{g} G^{-1}$. See [20] for explicit derivation. The operator that acts on U^a is the natural generalization of the Laplacian from flat spaces to manifolds, it is called the Laplace-Beltrami operator, and is denoted by Δ_g.

The parameter β, in the elements of the metric g_{ij}, determines the nature of the flow. At the limits, where $\beta \to 0$ and $\beta \to \infty$, we obtain respectively a linear diffusion flow and a nonlinear flow, akin to the TV flow for the case of grey-level images (see [21] for details).

The Beltrami scale-space emerges as a gradient descent minimization process

$$U^a_t = -\frac{1}{\sqrt{g}} \frac{\delta S}{\delta U^a} = \Delta_g U^a. \tag{3}$$

For Euclidean embedding, the functional in Eq. (1) reduces to

$$\begin{aligned} S(U) &= \int \sqrt{g}\, dx^1\, dx^2 \\ &= \int \sqrt{1 + \beta^2 \sum_{a=1}^{3} |\nabla U^a|^2 + \frac{1}{2}\beta^4 \sum_{a,b=1}^{3} |\nabla U^a \times \nabla U^b|^2}\, dx^1\, dx^2. \end{aligned}$$

This geometric measure can be used as a regularization term for color image processing. In the variational framework, the reconstructed image is the minimizer of a cost-functional. This functional can be written in the following general form,

$$\Psi = \frac{\alpha}{2} \sum_{a=1}^{3} ||KU^a - U_0^a||^2 + S(U),$$

where K is a bounded linear operator. In the denoising case, K is the identity operator and in the deblurring case, K is a convolution operator of U^a with a given filter. The parameter α controls the smoothness of the solution.

The modified Euler-Lagrange equations as a gradient descent process for each case are

$$U_t^a = -\frac{1}{\sqrt{g}} \frac{\delta \Psi}{\delta U^a} = -\frac{\alpha}{\sqrt{g}} (U^a - U_0^a) + \Delta_g U^a \quad \text{(denoising)}, \tag{4}$$

$$U_t^a = -\frac{1}{\sqrt{g}} \frac{\delta \Psi}{\delta U^a} = -\frac{\alpha}{\sqrt{g}} k(-x,-y) * (k * U^a - U_0^a) + \Delta_g U^a \quad \text{(deblurring)}, \tag{5}$$

where $Ku = k * u$ and k is often approximated by the Gaussian blurring kernel.

The above equations provide an adaptive smoothing mechanism. In areas with large gradients (edges), the fidelity term is suppressed and the regularizing term becomes dominant. At homogenous regions with low-gradient magnitude, the fidelity term takes over and controls the flow.

3 Standard Explicit Finite Difference Scheme

Our goal is to speed-up the convergence of the explicit scheme in Beltrami color processing. In this section, we detail the common explicit scheme. The applications we address are the Beltrami based smoothing, Beltrami based denoising, and Beltrami based deblurring.

We work on a rectangular grid with step sizes Δt in time and h in space. The spatial units are normalized such that $h = 1$. For each channel U^a, $a \in \{1, 2, 3\}$, we define the discrete approximation $(U^a)_{ij}^n$ by

$$(U^a)_{ij}^n \approx U^a(ih, jh, n\Delta t).$$

On the boundary of the image we impose the Neumann boundary condition.

The explicit finite difference scheme is written in a general form as

$$(U^a)^{n+1}_{ij} = (U^a)^n_{ij} + \mathit{\Delta} t O^n_{ij}(U^a), \tag{6}$$

where O^n_{ij} is the discretization of the right hand side of the relevant continuous equation (3), (4), or (5). Below, we give the exact form of the operator O^n_{ij} for each of the above cases.

– *Beltrami-based smoothing.*
  The explicit scheme (6) for discretizing Eq. (3) takes the form

$$(U^a)^{n+1}_{ij} = (U^a)^n_{ij} + \mathit{\Delta} t L^n_{ij}(U^a), \tag{7}$$

  where $L^n_{ij}(U^a)$ denotes a discretization of the Laplace-Beltrami operator $\mathit{\Delta}_g U^a$, for example, using a backward-forward approximation.
– *Beltrami-based denoising.*
  The explicit scheme (6) is given in this case by

$$(U^a)^{n+1}_{ij} = (U^a)^n_{ij} + \mathit{\Delta} t\big[L^n_{ij}(U^a) + \frac{\alpha}{\sqrt{g}}((U^a_0)^n_{ij} - (U^a)^n_{ij})\big]. \tag{8}$$

– *Beltrami-based deblurring.*
  Similarly, in the deblurring case, the explicit scheme (6) reads

$$(U^a)^{n+1}_{ij} = (U^a)^n_{ij} + \mathit{\Delta} t\big[L^n_{ij}(U^a) + \frac{\alpha}{\sqrt{g}}\bar{k}^n_{ij} * \big((U^a_0)^n_{ij} - k^n_{ij} * (U^a)^n_{ij}\big)\big], \tag{9}$$

  where $\bar{k} = k(-x, -y)$.

Due to stability requirements (see [5], [8]), explicit schemes limit the time-step $\mathit{\Delta} t$ and usually require a large number of iterations in order to converge. We propose to use vector extrapolation techniques in order to accelerate the convergence of these explicit schemes.

4 MPE/RRE Acceleration Techniques

Let us start by describing extrapolation methods for accelerating convergence of vector sequences. These techniques do not require information on the sequence generator, but are computed directly from the elements of the sequence. Following [16], we review the Minimal Polynomial Extrapolation (MPE) and Reduced Rank Extrapolation (RRE) methods.

These methods were first introduced for the case of linearly generated vector sequences (see [3], [6]) and were analyzed from the point of view of convergence, rate of convergence, and stability in [15]. In [17] these methods' convergence behavior was analyzed in the case of nonlinear problems . It is important to note that various related methods such as Krylov subspace methods and generalized conjugate residuals can be applied *only to linear systems*. Unlike these methods, the MPE and RRE techniques are applicable to nonlinearly generated sequences.

Let $\mathbf{x}_n, ..., \mathbf{x}_{n+k}$ be a given sequence of N dimensional column vectors, and denote by $\mathbf{s}_{n,k}$ its limit. These vectors are usually taken to be the last $k{+}1$ vectors of an iterative process. The vector $\mathbf{x}_n$ is not necessarily the initial solution, but rather can be taken at an arbitrary position in the sequence. In practice, after applying the acceleration technique and obtaining an approximate solution vector, this vector can be considered as the new $\mathbf{x}_n$.

The extrapolation methods have been derived based on differences, and below we use the abbreviated notation for first and second differences of the sequence of vectors. Denote by $\mathbf{u}_j$ and $\mathbf{w}_j$ the first and the second differences of the vectors $\mathbf{x}_i$.

$$\mathbf{u}_j = \mathbf{x}_{j+1} - \mathbf{x}_j, \quad \mathbf{w}_j = \mathbf{u}_{j+1} - \mathbf{u}_j, \quad j = 0, 1, 2, ... \tag{10}$$

Define the $N \times (j+1)$ matrices $\mathbf{U}_j^{(n)}$ and $\mathbf{W}_j^{(n)}$ by

$$\mathbf{U}_j^{(n)} = [\mathbf{u}_n | \mathbf{u}_{n+1} | \cdots | \mathbf{u}_{n+j}] \tag{11}$$

and

$$\mathbf{W}_j^{(n)} = [\mathbf{w}_n | \mathbf{w}_{n+1} | \cdots | \mathbf{w}_{n+j}], \tag{12}$$

respectively.

4.1 MPE Definition

Let k be a positive integer number. The approximation $\mathbf{s}_{n,k}$ to the desired limit $\mathbf{s}$ is given by

$$\mathbf{s}_{n,k} = \sum_{j=0}^{k} \gamma_j \mathbf{x}_{n+j}, \tag{13}$$

where the coefficients γ_j are determined as follows:

1. Obtain the least squares solution $\mathbf{c}$ for the overdetermined linear system

$$\mathbf{U}_{k-1}^{(n)} \mathbf{c} = -\mathbf{u}_{n+k},$$

   using the modified Gram-Schmidt algorithm [16].
2. Denote $\mathbf{c} = (c_0, c_1, ..., c_{k-1})^T$. Set $c_k = 1$ and compute the coefficients γ_j

$$\gamma_j = \frac{c_j}{\sum_{i=0}^{k} c_i}, 0 \le j \le k, \tag{14}$$

   assuming that $\sum_{i=0}^{k} c_i \neq 0$. When this condition is violated, $\mathbf{s}_{n,k}$ does not exist.

4.2 RRE Definition

For the RRE method, the approximation $\mathbf{s}_{n,k}$ of $\mathbf{s}$ is defined as

$$\mathbf{s}_{n,k} = \mathbf{x}_n + \sum_{i=0}^{k-1} \xi_i \mathbf{u}_{n+i}. \tag{15}$$

The coefficients ξ_i are determined by solving the overdetermined linear system

$$\mathbf{W}_{k-1}^{(n)} \xi = -\mathbf{u}_n, \tag{16}$$

with coefficients $\xi = (\xi_0, \xi_1, ..., \xi_{k-1})^T$.

Since there always exists a solution to the least square problem (16), $\mathbf{s}_{n,k}$ always exists. In particular, $\mathbf{s}_{n,k}$ exists uniquely when the matrix $\mathbf{W}_{k-1}^{(n)}$, has a full rank.

The essential difference between MPE and RRE is the way of determining the coefficients of the extrapolation. In our experiments, the MPE algorithm leads to visual results similar to the RRE, but converges more slowly than the RRE. Thus for the experiments displayed in the next section we chose the RRE as the extrapolation method.

Next, we describe the way of applying the vector extrapolation method in the context of the Beltrami framework. Assume the vector sequence generated by an explicit scheme is given by

$$\mathbf{x}_{n+1} = \mathbf{F}(\mathbf{x}_n), \tag{17}$$

where the nonlinear operator $\mathbf{F}$ is

$$\mathbf{F} = I + \Delta t O, \tag{18}$$

and O is given in Section 3 by either Eq. (7),(8), or (9).

The *vector extrapolation* method is then applied as follows:

1. We start with $n = 0$ and an initial guess $\mathbf{x}_0$, which is, in fact, the starting vector for the sequence to be generated. A few iterations of the explicit scheme (17) can be used as a preconditioner.
2. We use the explicit scheme (17) and $\mathbf{x}_n$ to generate a sequence of vectors $\mathbf{x}_n$,..., $\mathbf{x}_{n+k}$.
3. A vector extrapolation method (MPE/RRE) is used in order to extrapolate the approximation $\mathbf{s}_{n,k}$ of $\mathbf{s}$ from the last $k+1$ vectors, $\mathbf{x}_n$,..., $\mathbf{x}_{n+k}$.
4. The approximation $\mathbf{s}_{n,k}$ is used as a new starting point $\mathbf{x}_{n'}, n' = n + k + 1$ for a new sequence.
5. The procedure is repeated from Step 2 until convergence.

Sidi [16] used the term *cycling* to refer to this kind of scheme.

Note that the nonlinear eqs. (17), (18) describe the discretization of the Beltrami flow in the same form used by Smith et al. in the convergence analysis of nonlinearly generated sequences ([17], Section 6).

5 Experimental Results

We proceed to demonstrate experimental results of the Beltrami scale-space and restoration of color images processed by the explicit and the RRE accelerated schemes, specifying the CPU runtime and the resulting speed-up. Although in each application we display results with respect to a single image, the behavior exhibited is similar to other input data.

Let the sequence of vectors $\mathbf{x}_1, \mathbf{x}_2, ...$ be generated according to

$$\mathbf{x}_{n+1} = \mathbf{F}(\mathbf{x}_n), \quad n = 0, 1, ...$$

where $\mathbf{F}$ is given in (18). The *residual* for $\mathbf{x}_n$ is defined as

$$Res(\mathbf{x}_n) = \mathbf{F}(\mathbf{x}_n) - \mathbf{x}_n. \tag{19}$$

In our experiments we apply the RRE algorithm in the cycling mode (20 initial explicit iterations, and unless specified otherwise, $k = 10$). The RRE accelerated scheme allows us to reduce the number of explicit iterations by at least a factor of 10, in order to reach the same residual norm value. Experiments demonstrate that the RRE scheme remains stable as the number of iterations increases.

Figure 1 top row depicts the scale-space behavior of the Beltrami color flow, obtained using Eq. (7). At the bottom right it shows the speed-up obtained by using the RRE scheme for the Beltrami-based scale-space analysis. The speed-up gain is up to 40, as can be seen in the graph of the residual norm.

We measure the approximation error using l^2-norm values. Figure 2 shows the l^2-norm values of the images generated by the explicit and the RRE schemes during the scale-space evolution. Comparison is done by running the explicit scheme and the RRE scheme simultaneously. After each RRE iteration, we advance the explicit sequence, starting from the previous result until it diverges from the RRE result. l^2-norm values indicate that the images obtained by the explicit and RRE techniques are “numerically” the same. The maximum l^2-norm value observed during scale-space evolution was 0.194%. This validates numerically the convergence of the scheme.

5.1 Beltrami-based Denoising

Figure 3 displays the restoration of an image from its noisy version by applying Eq. (8). The speed-up in this case is about 10.

5.2 Beltrami-based Deblurring

In the next example the original image was blurred by a Gaussian kernel, as shown in Figure 4 top-left. The image was restored using Eq. (9). A significant speed-up is obtained in this case, as seen in Figure 4 bottom.

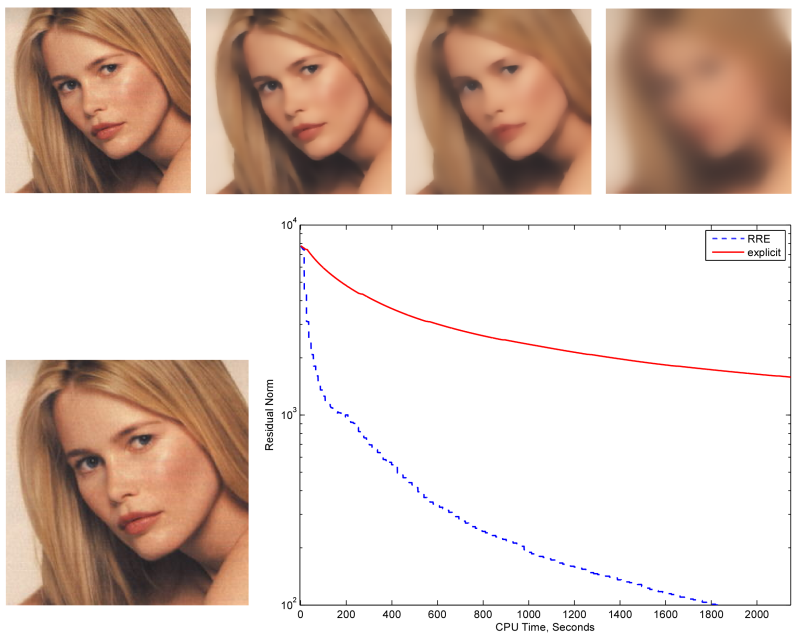


Fig. 1. Top (left to right): RRE iterations: $50, 150, 300, 450$. Bottom: left: Original picture. right: Comparison of the residual norms versus CPU time. Parameters: $\beta = \sqrt{1/0.0005} \simeq 44$, $\Delta t = 0.0042/\beta^2$.

6 Concluding Remarks

Due to its anisotropic nature and non-separability, Beltrami color flow discretizations are usually performed with explicit schemes. Low computational efficiency limits their use in practical applications. We accelerated the convergence of the explicit scheme using vector extrapolation methods. Experiments of denoising and deblurring color images based on the RRE algorithm have demonstrated the efficiency of the method. This makes vector extrapolation methods useful and attractive to the Beltrami filter and potentially other image processing applications.

Acknowledgment

We would like to thank Prof. Avram Sidi for his advice regarding vector extrapolation algorithms.

This research was supported in part by the European 6th framework program NOE-MUSCLE, in part by the Israeli Science Foundation (ISF) grant No. 738/04, and in part by the Horowitz fund.

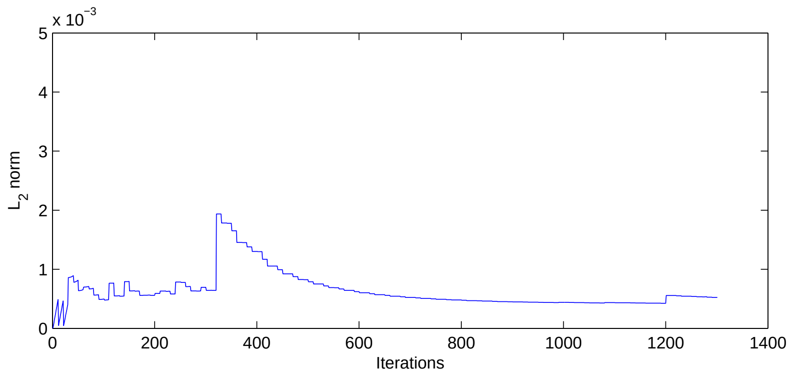


Fig. 2. l^2-norm error between the RRE iterations sequence and the corresponding explicit iterations images. A sequence using $k = 15$ is shown, which gave the worst l^2-norm values in practice. Parameters: $\beta = \sqrt{1/0.001} \simeq 31$, $\Delta t = 0.0042/\beta^2$.

References

1. L. Bar, A. Brook, N. Sochen, and N. Kiryati. Color image deblurring with impulsive noise. *Proceedings of the International Conference on Variational, Geometry and Level Sets Methods in Computer Vision (VLSM 05)*, 3752:49–60, 2005.
2. D. Barash. A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):844–847, 2002.
3. S. Cabay and L. Jackson. Polynomial extrapolation method for finding limits and antilimits of vector sequences. *SIAM Journal on Numerical Analysis*, 13(5): 734–752, 1976.
4. F. Catte, P. L. Lions, J. M. Morel, and T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis*, 29 (1):182–193, 1992.
5. R. Courant, K. Friedrichs, and H. Lewy. Uber die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, 100(1):32–74, 1928.
6. R. P. Eddy. Extrapolationg to the limit of a vector sequence. pages 387–396, 1979.
7. M. Elad. On the bilateral filter and ways to improve it. *IEEE Transactions On Image Processing*, 11(10):1141–1151, 2002.
8. B. Gustafsson, H. Kreiss, and J. Oliger. *Time dependent problems and difference methods*. Wiley, 1995.
9. R. Kimmel, R. Malladi, and N. Sochen. Images as embedding maps and minimal surfaces: Movies, color, texture, and volumetric medical images. *International Journal of Computer Vision*, 39(2):111–129, 2000.
10. T. Lu, P. Neittaanmaki, and X.-C. Tai. A parallel splitting up method and its application to Navier-Stokes equations. *Applied Mathematics Letters*, 4(2):25–29, 1991.
11. T. Lu, P. Neittaanmaki, and X.-C. Tai. A parallel splitting up method for partial diferential equations and its application to Navier-Stokes equations. *RAIRO Mathematical Modelling and Numerical Analysis*, 26(6):673–708, 1992.

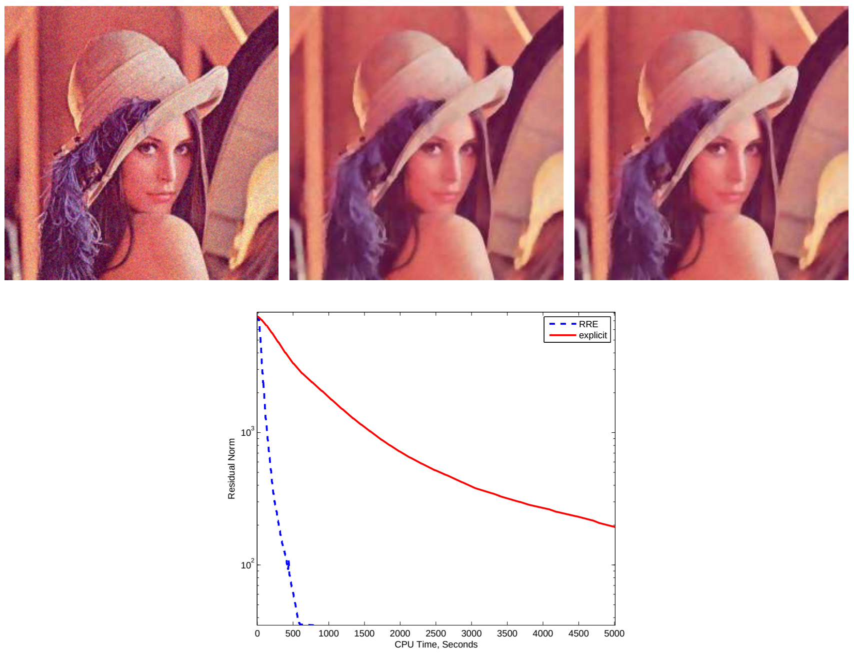
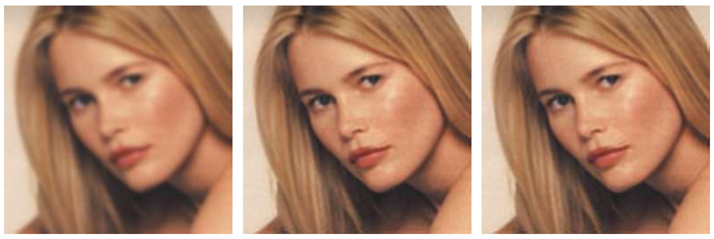


Fig. 3. Beltrami based denoising. Top left: Noisy image. Middle: Denoised image obtained by the RRE (901 iterations). Right: Denoised image obtained by the explicit scheme (11541 iterations). Bottom: Comparison of the residual norms versus CPU time. Parameters: $\beta = \sqrt{1000}$, $\lambda = 0.02$, $\Delta t = 0.0021/\beta^2$.

12. P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern Analysis and machine intelligence*, 12:629–639, 1990.
13. A. M. Polyakov. Quantum geometry of bosonic strings. *Physics Letters*, 103 B: 207–210, 1981.
14. L. Rudin, S. Osher, and E. Fatemi. Non linear total variation based noise removal algorithms. *Physica D Letters*, 60:259–268, 1992.
15. A. Sidi. Convergence and stability properties of minimal polynomial and reduced rank extrapolation algorithms. *SIAM Journal on Numerical Analysis*, 23(1):197–209, 1986.
16. A. Sidi. Efficient implementation of minimal polynomial and reduced rank extrapolation methods. *J. Comput. Appl. Math.*, 36(3):305–337, 1991.
17. D. A. Smith, W. F. Ford, and A. Sidi. Extrapolation methods for vector sequences. *SIAM Review*, 29(2):199–233, 1987.
18. S. M. Smith and J. Brady. Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23:45–78, 1997.
19. N. Sochen, R. Kimmel, and A. M. Bruckstein. Diffusions and confusions in signal and image processing. *Journal of Mathematical Imaging and Vision*, 14(3):195–209, 2001.



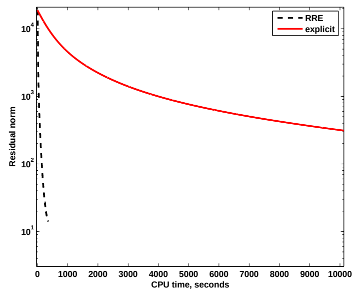


Fig. 4. Beltrami-based deblurring. Top left: Blurred image. Middle: Deblurred image obtained by the RRE scheme (1301 iterations). Right: Deblurred image obtained by the explicit scheme (196608 iterations). Bottom: Comparison of the residual norms versus CPU time. Parameters: $\beta = \sqrt{1/0.0005} \simeq 44$, $\Delta t = 0.0021/\beta^2$, $\lambda = 0.03$.

20. N. Sochen, R. Kimmel, and R. Maladi. From high energy physics to low level vision. *Proceedings of the First International Conference on Scale Space Theory in Computer Vision*, 1252:236–247, 1997.
21. N. Sochen, R. Kimmel, and R. Maladi. A general framework for low level vision. *IEEE Trans. on Image Processing*, 7:310–318, 1998.
22. A. Spira, N. Sochen, and R. Kimmel. Efficient Beltrami flow using a short time kernel. *Proceedings of Scale Space 2003, Lecture Notes in Computer Science*, 2695: 511–522, 2003.
23. C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *Proceedings of IEEE International Conference on Computer Vision*, pages 836–846, 1998.
24. J. Weickert. Efficient and relible schemes for nonlinear diffusion filtering. *IEEE Trans. on Image Processing*, 7:398–410, 1998.
25. J. Weickert. Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31:111–127, 1999.
26. A. J. Yezzi. Modified curvature motion for image smoothing and enhancement. *IEEE Transactions on Image Processing*, 7(3):345–352, 1998.