Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Biomedical Informatics 35 (2002) 77–91

Journal of Biomedical Informatics

www.academicpress.com

Evolutionary partial differential equations for biomedical image processing

Alessandro Sarti,[a,b,*] Karol Mikula,[c] Fiorella Sgallari,[d] and Claudio Lamberti[b]

[a] *Department of Mathematics, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA, USA*
[b] *DEIS, University of Bologna, Viale Risorgimento, 2 I 40136 Bologna, Italy*
[c] *Department of Mathematics, Slovak University of Technology, Radlinskeho 11, 813 68 Bratislava, Slovakia*
[d] *Department of Mathematics, University of Bologna, Italy*

Received 2 November 2001

Abstract

We are presenting here a model for processing space–time image sequences and applying them to 3D echo–cardiography. The non-linear evolutionary equations filter the sequence with keeping space–time coherent structures. They have been developed using ideas of regularized Perona–Malik an-isotropic diffusion and geometrical diffusion of mean curvature flow type (Malladi–Sethian), combined with Galilean invariant movie multi-scale analysis of Alvarez et al. A discretization of space–time filtering equations by means of finite volume method is discussed in detail. Computational results in processing of 3D echo–cardiographic sequences obtained by rotational acquisition technique and by real-time 3D echo volumetrics acquisition technique are presented. Quantitative error estimation is also provided.

Keywords: Medical imaging; Echo–cardiography; Motion based image filtering; Image multi-scale analysis; Non-linear partial differential equations; Numerical solution

1. Introduction

The aim of this contribution is to present a mathematical model and numerical method in the processing of three-dimensional (3D) image sequences. The proposed model is applied to 3D and 3D + time echo–cardiography. The model used for space–time filtering is based on partial differential equations (PDEs) approach, that is PDEs of degenerate diffusion type are applied to initially given image sequence. Non-linear PDEs for the processing of this kind of data have been used also in [1–9]. Since the images are given on discrete grids, in this study the nonlinear PDEs are discretized by semi-implicit finite volume method to get a fast and stable solution.

Two-dimensional (2D) echo–cardiography is an imaging modality frequently used in cardiology due to its simplicity, lack of ionizing radiation, and a relative low cost. However, 2D echo–cardiography allows visualization of only tomographic planar sections of the heart; thus to obtain a complete evaluation of the heart anatomy and function, the physician must reassemble mentally a 3D model from multiple 2D images. Moreover, 2D echo–cardiography relies on geometrical assumptions for the determination of heart chamber volumes and thus presents a considerable measurement error, especially for the right ventricular and the atrial volume determination [10]. Three-dimensional echo–cardiography may avoid the need for geometrical assumptions, thereby allowing accurate evaluation of the size and shape of the chambers, even in the case of cavities with irregular or distorted geometry. The correct visualization and interpretation of 3D echo images is often affected by the high amount of noise intrinsically linked to the acquisition method. It is absolutely necessary to submit the data to pre-processing to improve their legibility from a clinical point of view. The application of traditional pre-processing algorithms (moving average, median, and Gaussian filtering) does not reduce the noise superimposed to the image maintaining a good

[*] Corresponding author. Fax: +39-051-2093073.
E-mail address: asarti@deis.unibo.it (A. Sarti).

1532-0464/02/$ - see front matter © 2002 Elsevier Science (USA). All rights reserved.
PII: S1532-0464(02)00502-6

definition of the interfaces. The importance of noise removal by non-linear filtering both for visualization purposes and optical flow estimation has been also outlined by several authors (see e.g., [11,12]). The processing algorithm should be able to distinguish the noise from the contours of the different cardiac structures by using both spatial and temporal coherence.

In this paper, we will use two types of 3D echo–cardiographic data sets to which our algorithms are applied and tested. The first sequence has been obtained by means of a rotational acquisition technique using the TomTec Imaging System [13]. With this technique the transducer undergoes a rotation around its main axis in a propeller configuration. A series of tomographies corresponding to the sections of a cone of biological tissue has been acquired. The acquisition consists of 14 image-cubes representing a whole cardiac cycle of a real patient. A volume of interest of 150 × 150 × 100 voxels will be processed. The time interval between one cube and the next one is 40 ms. Figs. 2–5, 7–13 are related to results on this data set. The quality of this 3D raw dataset is quite good. Nevertheless a remarkable amount of noise is present in the sequence and thus it is a reasonable testing example for the proposed methods.

The second type of processed data is given by real-time 3D ultrasound echo-images. Real-time 3D echo (RT3DE volumetrics) acquisition technique [14,15] is characterized by a 43 × 43 piezoelectric elements transducer with 2.5–3.5 MHz frequency, 60 × 60 degree angular opening and 256 transmission lines (see Fig. 1). There are 64 × 64 pixels in each of 512 C-scan planes and 2.097.152 voxels in every 3D frame. The main difference between RT3DE and 2D rotational acquisition is that RT3DE can provide images of left ventricle without ECG/respiratory gating, there is less acquisition time and no processing time. RT3DE can also be used in arrhythmia cases and no special training is required. However, there is less resolution because of broadened transmit pattern.

From a mathematical point of view, the input image sequence, representing an acquisition of moving objects, can be modelled by a real function $u^0(x,\theta)$, $u^0 : \Omega x$ $[0,T] \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^N$ represents a spatial domain, $x = (x_1,\ldots,x_N)$ represents a spatial point and θ a point in the time interval $[0,T]$ in which acquisition is realized. In practice, Ω is a rectangular domain, $N = 2$ or 3 and, in special applications, the time sequence can be periodically prolonged from $[0,T]$ to $\mathbb{R}$. The typical example which can be represented by such u^0 is an ultrasound acquisition of beating heart in 3D echo–cardiography (see numerical examples in following sections).

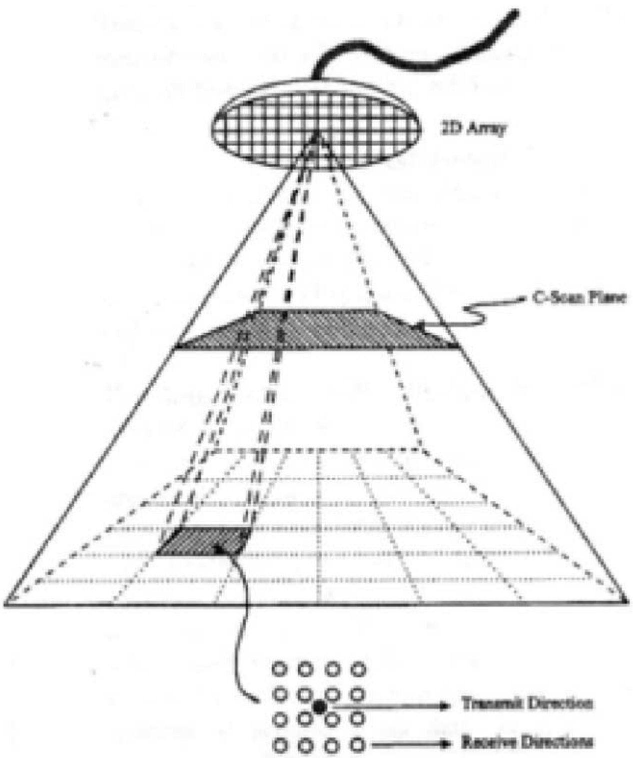

Fig. 1. Real-time 3D echo (RT3DE volumetrics) acquisition technique.

The application of PDE to initially given (noisy) image sequence can be understood as its embedding to the so-called *non-linear scale space* [16–19]. The axioms and fundamental properties of such embedding has been given and studied in [16] and the notion of *image multi-scale analysis* has been introduced. The multi-scale image analysis associates with u^0 a family $u(t,x,\theta)$ of smoothed-simplified images (in our case a family of smoothed sequences) depending on an abstract parameter t, the *scale*. As has been proved in [16], if such family fulfils basic assumptions—pyramidal structure, regularity and local comparison principle—then $u(t,x,\theta)$, $u : [0,T_s] \times \Omega \times [0,T] \to \mathbb{R}$ can be represented as unique viscosity solution (in the sense of [20]) of a second-order (degenerate) parabolic partial differential equation

$$\frac{\partial u}{\partial t} = F(t,u,\mathrm{D}u,\mathrm{D}^2u), \tag{1}$$

with the initial condition given by

$$u(0,x,\vartheta) = u^0(x,\vartheta), \tag{2}$$

where $\mathrm{D}u$ denotes a vector of first partial derivatives with respect to all space and time variables $x_1,\ldots,x_N$, θ, by ∇u we denote its spatial part, D^2u denotes a matrix of second-order derivatives with respect to all space and time variables and F in (1) is a non-decreasing function of the fourth argument.

The equations of (degenerate) parabolic type have a smoothing property, so they are natural tools for filtering (image simplification) by removing spurious structures, e.g., a noise. However, the simplification should be "image oriented," e.g., it should respect edges and not blur them. Or, it should recognize motion of a structure in image sequences and consequently the smoothing (diffusion) should respect the motion coherence in subsequent frames. Such, or even more sophisticated requirements related to geometrical characteristics of image, bring strong non-linearity into the parabolic PDEs (diffusion can depend on $|\nabla u|$—edge indicator) or even degeneracy (diffusion can be stopped in points which are "un-noisy" by motion field information). Hence one can see that PDEs used for processing of image sequences, or static frames of the sequence, must cover complex situations and handle wide range of phenomena. In the next sections, some models are presented which are suitable for such goals and review also partial approaches for filtering of 2D and 3D images, frames of the sequence.

In Section 2, some basic concepts of geometrical diffusion of mean curvature flow type are introduced. We discuss the main features related to image selective smoothing and give references to works describing computational methods based on finite element and finite volume approximations for solving numerically given non-linear parabolic equations. Concerning details of variational numerical methods applied in image processing we refer to review paper [21]. Some computational experiments with medical images are also included to illustrate application of the models.

In Section 3, we will present a model for space–time filtering combining spatial diffusion equations from Section 2 and motion coherence of moving objects in time. In Section 4, we will discuss in detail numerical methods based on finite volume approximation for solving our space–time filtering equations. In Section 5, we will present computational results obtained by such schemes in filtering of artificial as well as echo–cardiographic image sequences.

2. Geometrical diffusion for space filtering of 2D and 3D images

In this section, a model of surface evolution used for processing frames of the image sequence independently of each other is considered. In the next chapter, such spatial smoothing processes will be combined with motion coherence in entire image sequence. We will denote $v^0(x) = u(0, x, \theta^*)$ the frame of initially given image sequence at some moment $\theta^* \in [0, T]$. In [1,2] an intrinsic diffusion equation to 3D echo–cardiographic frames is applied. We call it geometrical diffusion since it is related to geometrical (or intrinsic) diffusion equations on manifolds.

In rather general situations, the blood–cardiac muscle interface corresponds to an iso-surface (iso-line in 2D) of the grey-level image intensity function and hence it forms a recognizable silhouette in the image. This phenomenon is clearly visible in Fig. 2. To remove the unsmoothness of the silhouette (in the original image or in the pre-filtered by anisotropic diffusion image) caused by errors in acquisition, it seems reasonable to move such iso-surface (iso-curve) in direction of its normal vector field with a velocity proportional to the mean curvature. The motions of convex and concave pieces are opposite due to the curvature sign, and the large fingers shrink much faster than smoother parts due to the curvature dependence of the flow. Thus, locally in scale, we can obtain desirable smoothing of the silhouette.

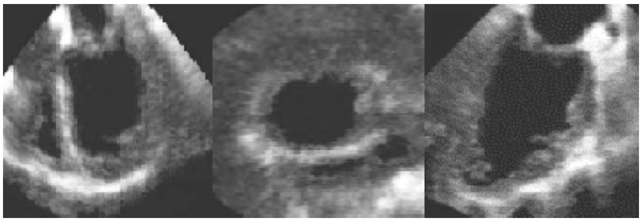

Fig. 2. Two-dimensional cuts of 3D echo–cardiographic cube with image of the human left ventricle.

This approach is nothing else than a curvature driven evolution of iso-curve or iso-surface of the image. Such problems are well known in a lot of applications related to the so-called free-boundary problems. From a computational point of view there are two main approaches to solve numerically a curve or surface evolution governed by curvature. In the first, so-called *Lagrangean approach* [22–25], the curve or surface itself is discretized. Then a system of equations is derived (e.g., by semi-implicit finite element or finite difference methods) for the new position of discrete curve or surface representation points. This system is solved to get new iso-surface (iso-curve) representation. The Lagrangean approach is a very efficient and computationally fast method, but it can hardly handle the evolution through singularities, splitting, and merging of the curves or surfaces, respectively, during the evolution. In spite of this, the so-called *Eulerian approach* implicitly handles the curvature driven motion passing the problem to higher dimensional space and solving evolutionary equation for a graph time evolving level sets of which correspond to an evolving curve or surface. The *phase field equations* (see e.g., [26] and *level set methods* [27–31] are approaches of that type. Especially Osher and Sethian proposed the *level set equation* for curvature dependent motion

$$\frac{\partial v}{\partial t} = |\nabla v| \nabla \cdot \left(\frac{\nabla v}{|\nabla v|} \right), \tag{3}$$

where $v(t, x)$ is the unknown function defined in $Q_{T_s} \equiv [0, T_s] \times \Omega$. We assume that $\Omega \subset \mathbb{R}^N$ is a bounded rectangular domain, $[0, T_s]$ being the scaling interval. This model is well suited for image processing applications, since all geometrical information about image level lines or level surfaces is handled in once. The level set equation not only moves each level set of v with a velocity proportional to its normal mean curvature field but it also fulfills the so called *morphological principle*; if v is a solution, then for any non-decreasing function ϕ, $\phi(v)$ is a solution as well. This *contrast invariant* property has large significance in the theory of image processing [16] (even if not in all application it is desirable, because of its noise sensitivity). Such property also means that all level sets move independently on each other, they diffuse only intrinsically and there is no diffusion across them in the normal direction. This idea was used in [32], where equation

$$v_t - g(|\nabla G_\sigma * v|)|\nabla v|\nabla \cdot \left(\frac{\nabla v}{|\nabla v|}\right) = 0 \qquad (4)$$

has been suggested for computational image and shape analysis, where $g : \mathbb{R}_0^+ \to \mathbb{R}^+$ is a non-increasing smooth function, $g(0) = 1$, $g(s) \to 0$ for $s \to \infty$. In Eq. (4) $G_\sigma \cup C^\infty(\mathbb{R}^N)$ is a smoothing kernel (e.g., Gauss function) with unit mass and tending to Dirac function as $\sigma \to 0$. The convolution in (4) is understood in the usual sense

$$\nabla G_\sigma * v = \int_{\mathbb{R}^N} \nabla G_\sigma(x - \xi)\tilde{v}(\xi)\,\mathrm{d}\xi, \qquad (5)$$

where $\tilde{v}$ is an extension of v to $\mathbb{R}^N$ given by a periodic reflection through the boundary of Ω. Eq. (4) is accompanied with zero Neumann boundary conditions and initial condition

$$\frac{\partial v}{\partial \nu} = 0, \text{ on } [0, T_s] \times \partial\Omega, \qquad (6)$$

$$v(0,x) = v^0 \text{ in } \Omega, \quad v^0 \in L_\infty(\Omega), \qquad (7)$$

where ν is unit normal vector to the boundary of Ω.

From a practical point of view, applying just the level set equation (i.e., when $g \equiv 1$) to the initial image yields the intrinsic silhouettes smoothing [1,2]. On the other hand, Eq. (4) can be used successively for image selective smoothing with preserving edge positions. The function $g(s)$ depending on $|\nabla G_\sigma * v|$ is used to slow down the motion of the silhouettes which are at the same time unspurious edges. The regions between them are smoothed by the mean curvature flow. In case of geometrical diffusion we again consider zero Neumann boundary conditions on $\delta\Omega$ and initial condition given by processed image v^0. A 2D and 3D filtering algorithm has been also proposed in [4] where the Eq. (3) has been rewritten with respect to a Riemannian metric induced by the image.

We will present an application of geometrical diffusion to medical image processing. The *level set* Eq. (3) is degenerate parabolic and hence rather complicated from the numerical point of view. Its viscosity solution [33–35] can be tracked numerically e.g., by the well-known level set method based on a solution of first-order Hamilton–Jacobi equation [27,28].

However, in curvature driven motion one can use also another approach leading to usage of standard numerical methods for solving parabolic PDEs, namely, finite element or finite volume methods for discretization in space and semi-implicit method in scale. The semi-implicit schemes [9,35–42], where nonlinear terms of equation are treated from the previous discrete scale step and linear terms are considered on the current scale level, have favourable stability and efficiency properties and they converge to the weak solution of the parabolic problem. Such approach is $L\,\infty$-stable and leads to solving linear systems in every discrete scale level. For space discretization either finite element method [9,36,37,42], finite volume method [38–40] or finite difference method [35,41], can be used. Hence one can use state-of-the-art methods of numerical linear algebra and pre-conditioners. For details of implementation by means of variational methods we refer e.g., to [21]. For general overview of modelling and further numerical approaches one can also consult recent books or paper collections [43–46].

We will try to obtain a realistic—smooth shape of the left ventricle of the heart. First, in Fig. 3 the smoothing effect of geometrical diffusion (3) is presented. On the left side the cuts of unfiltered iso-surfaces are plotted, on the right the filtered ones are (both after binarization with the same treshold). The extinction of small structures due to their high curvature and smoothing of the important structures can be seen in the image. In Figs. 4 and 5, we visualize the level surfaces which represent the

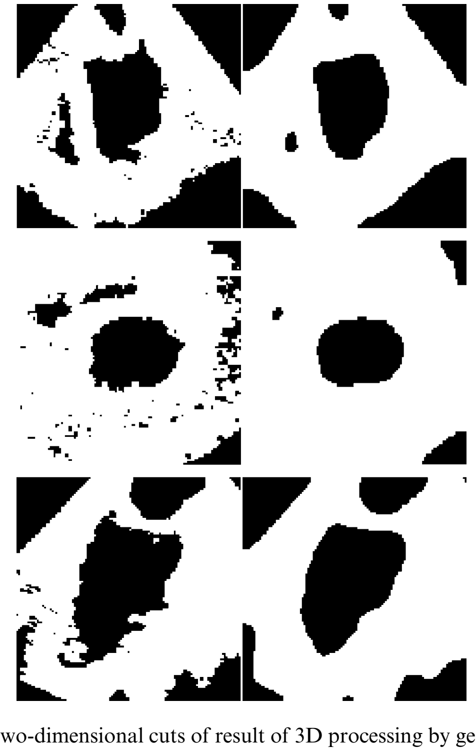

Fig. 3. Two-dimensional cuts of result of 3D processing by geometrical diffusion of mean curvature flow type.

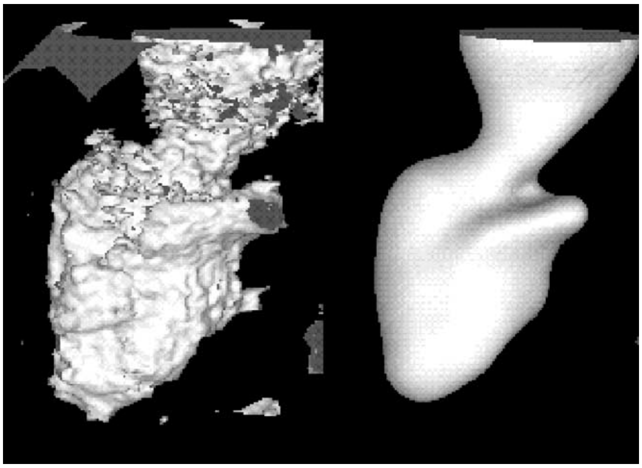

Fig. 4. Ventricular shape extraction using geometrical diffusion.

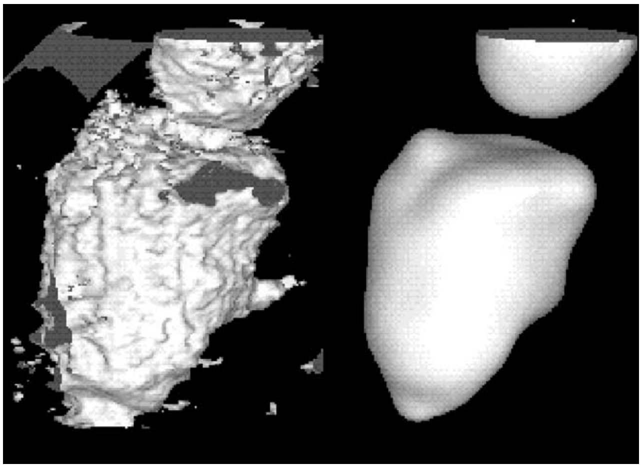

Fig. 5. Ventricular shape extraction using geometrical diffusion.

boundary of the volume containing the blood in two discrete moments of cardiac cycle from the TomTec testing data set (processing of 5th and 14th time steps of sequence is presented). On the left side of the figures, the unfiltered iso-surfaces are plotted. The computational results are plotted on the right sides. In the presented numerical experiments, we have computed 21 discrete scale steps on Cray C92 and then stopped the geometrical diffusion. For longer time-scale the diffusion tends to shrink the structure. This shortcoming of the method will be improved by considering space–time coherence of the entire sequence given in the following section.

At the end of this section, let us mention a general useful view to gradient dependent non-linear (geometrical) diffusion equations. The non-linear diffusion term $\nabla \cdot (g(|\nabla u|)\nabla u)$ can be rewritten in 2D as

$$\nabla \cdot (g(|\nabla u|)\nabla u) = g(|\nabla u|)u_{\xi\xi} + H'(|\nabla u|)u_{\eta\eta}, \tag{8}$$

where $H(s) = sg(s)$ and ξ, η are tangential and orthogonal vectors to the level line, respectively. From this form, one can simply see how diffusion works along and across the image silhouettes with different choices of g. There is always positive, but possibly slowed-down (depending on the shape of g) diffusion along level lines. Across level line there can be forward diffusion (when $H'(s)$ is positive), zero diffusion (e.g., in Rudin–Osher–Fatemi model [47] dealing with total variation de-noising and also in the mean curvature flow equation in the level set formulation) or backward diffusion (in the original Perona–Malik model [17]).

3. Space–time multiscale filtering

A 3D space–time image sequence (e.g., in 3D echocardiography) $u^0(x,\theta)$ is a 4D image and we can apply a general multiscale analysis model (1) and (2) to it. The question is how to choose right-hand side of (1) with the aim at extracting relevant information from the sequence, filtering out the noise and enhancing moving structures. To that goal, we would like to use an additional information (in comparison with static image processing) given by motion coherence in the image sequence. We will assume that certain objects acquired at different times, and thus being in different frames of the sequence, are formed by points that preserve their intensity along the motion trajectory. Such objects are called Lambertian structures. Moreover we will assume that motion is Galilean locally in time, i.e., the motion trajectories are smooth. Designing the model equations, we will consider the following quantity [17,50] proposed by Guichard [51]

$$\begin{aligned}\mathrm{clt}(u) = \min_{w_1,w_2} \frac{1}{(\Delta\vartheta)^2}(&|\langle\nabla u, w_1 - w_2\rangle| \\ &+ |u(t,x-w_1,\vartheta-\Delta\vartheta) - u(t,x,\vartheta)| \\ &+ |u(t,x+w_2,\vartheta+\Delta\vartheta) - u(t,x,\vartheta)|),\end{aligned} \tag{9}$$

where w_1, w_2 are arbitrary vectors in $\mathbb{R}^N$ and $\Delta\theta$ is the increment in time. The scalar function $\mathrm{clt}(u)$ (the name clt indicates the relation to the curvature of Lambertian trajectory) will introduce a measure of coherence in time for the moving structures. It consists of the sum of three positive parts and we want to find the minimum in all possible directions w_1, w_2. The last two terms in the sum on the right-hand side of (11) are related to the differences in the intensities of end-points of candidate Lambertian velocity vectors w_1 and w_2. To find the directions of such vectors we have to look at the points which have the closest intensity value to the intensity $u(t,x,\theta)$ in the previous frame (term $|u(t,x-w_1,\theta-\Delta\theta) - u(t,x,\theta)|$) and in the next frame (term $|u(t,x+w_2, + \Delta\theta) - u(t,x,\theta)|$). Those differences are scaled by the factor $1/(\Delta\theta)^2$. Note that, if we find corresponding Lambertian points both terms vanish. The first term in the sum, namely $|\langle\nabla u, w_1 - w_2\rangle|/(\Delta\theta)^2$ corresponds to the so-called *apparent acceleration*, i.e. to the difference between candidate Lambertian velocity vectors w_1 and w_2 in the direction of ∇u. For details and more background from the optical flow point of view refer to [16,48,49]. The quantity $\mathrm{clt}(u)$ is thus related to the curvature of the space–time level curve passing through the space–time point (x,θ) in the scale t. The value of $\mathrm{clt}(u)$ vanishes for the Lambertian points that are in Galilean motion. It is consistent with our purposes not to alter such trajectories. On the other hand for the noisy points there is no motion coherence and thus $\mathrm{clt}(u)$ will be large there.

Concerning the space coherence, we assume that distinguished structures are represented as uniform regions of the image intensity function and that object boundary forms an edge in the image. To construct spatial diffusion process we will require specific behavior on the edges, e.g., it is desirable not to blur them, to keep their position as fixed as possible.

Another reasonable choice can be related to a smoothing of the edges by intrinsic diffusion (for that goal, flow by mean curvature can be used). There are diffusion processes designed to respect such features (see Section 2); we can choose Perona–Malik like anisotropic diffusion [17], mean curvature flow of curves and surfaces in the level set fomulation, as introduced in chapter 2, models based on minimization of image total variation, etc.

To combine time coherence of moving objects with their spatial localization we will consider the following equation

$$\frac{\partial u}{\partial t} = \mathrm{clt}(u)sd(u), \tag{10}$$

where spatial diffusion is given either by the Perona–Malik term, i.e.,

$$sd(u) = \nabla \cdot (g(|\nabla u|)\nabla u) \tag{11}$$

or by level set like term, i.e.,

$$sd(u) = |\nabla u|\nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right). \tag{12}$$

To prevent possible degeneracies we will regularize the equations, and usually use function g as

$$g(s) = \frac{1}{1+Ks^2} \tag{13}$$

with some constant K. To prevent ill-posedness of equation in such case, we use spatial regularization by smoothing kernel due to [35]. In case of Perona–Malik spatial smoothing we have proposed the following equation for the processing of image sequences [3]

$$\frac{\partial u}{\partial t} = \mathrm{clt}(u)\nabla \cdot (g(|\nabla G_\sigma * u|)\nabla u) \tag{14}$$

together with initial condition (2). The practical choice of the kernel is N-dimensional Gauss function

$$G_\sigma(x) = \frac{1}{(2\sqrt{\pi\sigma})^N}\mathrm{e}^{-|x|^2/4\sigma}. \tag{15}$$

Given (15), the term $\nabla G_\sigma \ * u$ is nothing else than the gradient of the solution at time σ of the linear heat equation starting from u initially. Thus, in Eq. (14), the spatial diffusion process is slowed down in the points in which the Gaussian gradient $\nabla G_\sigma \ * u$ is large (i.e., on edges) in spite of the regions with the constant signal mean (there, the smoothing is close to usual linear diffusion due to $g(0) = 1$). Eq. (14) preserves moving in time structures as well as keeping their spatial edges.

In the second case (12), i.e., when we are interested in the smoothing of moving object silhouette by intrinsic diffusion we use regularization in the sense of Evans and Spruck and thus consider the equation

$$\frac{\partial u}{\partial t} = \mathrm{clt}(u)\sqrt{\varepsilon^2 + |\nabla u|^2}\nabla \cdot \left(\frac{\nabla u}{\sqrt{\varepsilon^2+|\nabla u|^2}}\right), \tag{16}$$

where ε is a small regularization parameter. Eq. (16) is again coupled with initial condition (2). In both cases, (14) and (16), we consider zero Neumann boundary conditions in the spatial part of boundary and e.g., periodic boundary conditions in time boundary of the sequence.

In models (14) and (16), the change of image intensity in scale, i.e., $\delta u|\delta t$, is given by the right-hand side of (14) or (16). There the spatial diffusion term is multiplied by $\mathrm{clt}(u)$. Thus, the diffusion process degenerates (is stopped) in the Lambertian points that are in Galilean motion. This is the important difference from standard selective smoothing processes for static images. We can conclude that Eqs. (14) and (16) preserve moving in time structures as well as keep (or slightly smooth) their spatial edges. The features of the spatial non-linear selective smoothing and the Guichard's acceleration term $\mathrm{clt}(u)$ are connected.

In the next section we will present discretization and numerical solution of the partial differential Eqs. (14) and (16) by means of a finite volume method. Then we will discuss the solution of arising linear systems by means of iterative methods and give computational results with real and artificial images.

4. Numerical algorithm

In this section, a method for numerical solution of image sequence multiscale analysis Eqs. (14) and (16) is described. First let us give some notations. Let our space–time sequence consist of $m+1$ frames. Let $\theta = \Delta\theta = T/m$ be a discrete time step of the sequence. Let us denote a discrete scale step by τ. Then by u_j^i we denote the jth frame of the sequence in ith discrete scale step, i.e.,

$$u^i_j(x_1,x_2,x_3) = u(i\tau, x_1, x_2, x_3, j\vartheta). \tag{17}$$

The basic idea of our numerical method is to handle terms in (14) or (16) in such a way as to obtain a linear boundary value problem for u^i_j. The reason is that such equations can be solved by robust and efficient spatial discretization techniques based on finite volume (FVM), finite difference (FDM), or finite element methods (FEM). To that goal, the nonlinearities of Eq. (16) are treated using the previous scale step, while the linear terms are handled implicitly. Such approach is called semi-implicit approximation in scale. Then we provide space–time discretization and, finally, our numerical method leads to solving linear algebraic systems to update each frame of the sequence in a new scale.

Let us discuss the discretization of the terms in (14) in detail. From definition (9), we obtain a time-discrete version of $\mathrm{clt}(u)$ considering current, previous and next time frame of the sequence. So, we define

$$\mathrm{clt}(u^i_j) = \min_{w_1,w_2}\frac{1}{\vartheta^2}\Big(|\langle\nabla u^i_j, w_1 - w_2\rangle| + |u^i_{j-1}(x - w_1) - u^i_j(x)| + |u^i_{j+1}(x + w_2) - u^i_j(x)|\Big). \tag{18}$$

Now, we can write the *semi-implicit scheme* for solving (14).

Let τ and σ be given outer and inner discrete scale steps. For $i = 1, 2, \ldots$ and for each frame $j = 0, \ldots, m$, we look for u^i_j fulfilling

$$\frac{u^\sigma_j - u^{i-1}_j}{\tau} = \mathrm{clt}(u^{i-1}_j)\nabla\cdot\Big(g(|\nabla G_\sigma * u^{i-1}_j|)\nabla u^i_j\Big), \tag{19}$$

where the periodicity in time of the sequence is used for $j = 0$ and $j = m$ and the zero Neumann boundary conditions are considered for spatial boundary Ω. Let us mention that we can also use other conditions for updating the first and last frame in the sequence, e.g., reflexive if we have given only one half of the periodic cycle, or the first and last frame can serve as Dirichlet data for computing of u^i_j, $j = 1$, $j = m - 1$. Eq. (19) is elliptic PDE in points where $\mathrm{clt}(u^{i-1}_j) > 0$ while it degenerates to algebraic identity in points where $\mathrm{clt}(u^{i-1}_j) = 0$.

Given (15), we can realize the convolution, involved in computing of diffusion coefficient $g(|\nabla \mathrm{G}_\sigma * u^{i-1}_j|)$, by solving numerically the linear heat equation

$$\frac{\partial w}{\partial t} = \nabla\cdot(\nabla w) \tag{20}$$

in time interval $[0, \sigma]$ with the initial condition $w(x, 0) = u^{i-1}_j(x)$. Then we put $u^\sigma_j := w(x, \sigma)$. Numerically, we solve the Eq. (20) implicitly in t with just one discrete inner scale step with the length σ. This way, (19) can be rewritten into the couple

$$\frac{u^i_j - u^{i-1}_j}{\tau} = \mathrm{clt}(u^{i-1}_j)\nabla\cdot\Big(g(|\nabla u^\sigma_j|)\nabla u^i_j\Big), \tag{21}$$

where u^σ_j is the solution of

$$\frac{u^i_j - u^{i-1}_j}{\sigma} = \nabla\cdot(\nabla u^\sigma_j) \quad j = 0, \ldots, m, \quad i = 1, 2, \ldots \tag{22}$$

Now we will discuss the solution of (21) and (22) on given spatial discrete pixel/voxel structure. In discrete settings, $\min_{w_1,w_2}$ in (18) is only evaluated for vectors connecting pixel/voxel centers P also representing the nodes of the computational grid (see also [49]. In practice, we only consider a certain (not too large) rectangular neighbourhood centered in P. For space discretization of (21) as well as (22) we use, in this work, the so called finite (or control) volume method. This method is widely used in the engineering and applied mathematics community for numerical computations of diffusion (convection–diffusion and conservation laws as well) processes [48]. Next, we provide the discretization of Eq. (21). Case (22) is left to readers due to simplicity.

Let the discrete image intensity values be given in central points P of finite volumes corresponding to voxels in 3D. Let the distance between two of such points be h (we consider uniform 3D grid in this explanation, but all considerations can be generalized to nonuniform, nonrectangular 2D or 3D grids). Let us denote the grid neighbours of P by W (west), E (east), S (south), N (north), B (bottom), U (up) and the points crossing the finite volume in the direction of neighbours by w, e, s, n, b, and u. The finite volume around P then can be written as

$$V = [w, e] \times [s, n] \times [b, u] \subset \mathbb{R}^3.$$

The finite volume method consists in stating the assumptions on profiles of data and unknowns in various terms of equation and then integrating the equation through finite volume V. The general rule is used, in each term we assume such profile of functions which is sufficient to provide the differentiations if they are involved. We can proceed in such a way, because the final output consists just on values of unknown in discrete points and we can take any interpolating profile of the solution in between the grid points.

To simplify the notation, denote $d \equiv g(|\nabla u^\sigma_j|)$. Let us integrate Eq. (23) in finite volume V. We obtain

$$\begin{aligned}\int_w^e\int_s^n\int_b^u u^i_j\,\mathrm{d}x_1\,\mathrm{d}x_2\,\mathrm{d}x_3 &= \tau\int_w^e\int_s^n\int_b^u \mathrm{clt}(u^{i-1}_j)\left(\frac{\partial}{\partial x_1}\left(d\frac{\partial u^i_j}{\partial x_1}\right)\right.\\ &\quad\left.+\frac{\partial}{\partial x_2}\left(d\frac{\partial u^i_j}{\partial x_2}\right)+\frac{\partial}{\partial x_3}\left(d\frac{\partial u^i_j}{\partial x_3}\right)\right)\mathrm{d}x_1\,\mathrm{d}x_2\,\mathrm{d}x_3\\ &\quad+\int_w^e\int_s^n\int_b^u u^i_j\,\mathrm{d}x_1\,\mathrm{d}x_2\,\mathrm{d}x_3.\end{aligned} \tag{23}$$

Let $\mathrm{clt}(u^{i-1}_j)(x) \equiv \mathrm{clt}(u^{i-1}_j)(P)$ for all $x \in V$, i.e., we assume constant profile of this quantity in V. Further, assume that diffusion fluxes through boundaries of V are constant on each side of volume V; of course, in such case, they are given by the values in crossing points mentioned above. Then, using Newton–Leibnitz formula in first integral on the right-hand side of (23) we obtain

$$
\begin{aligned}
&\int_w^e \int_s^n \int_b^u u_j^i \,\mathrm{d}x_1\,\mathrm{d}x_2\,\mathrm{d}x_3 \\
&\quad = \tau \operatorname{clt}(u_j^{i-1})(P)h^2\bigg(d(e)\frac{\partial u_j^i(e)}{\partial x_1} - d(w)\frac{\partial u_j^i(w)}{\partial x_1} \\
&\qquad + d(n)\frac{\partial u_j^i(n)}{\partial x_2} - d(s)\frac{\partial u_j^i(s)}{\partial x_2} + d(u)\frac{\partial u_j^i(u)}{\partial x_3} \\
&\qquad - d(b)\frac{\partial u_j^i(b)}{\partial x_1}\bigg) + \int_w^e \int_s^n \int_b^u u_j^{i-1}\,\mathrm{d}x_1\,\mathrm{d}x_2\,\mathrm{d}x_3. \qquad (24)
\end{aligned}
$$

Approximating partial derivatives on the boundaries of finite volume by central differences and approximating integrals using values at central points, we obtain the following difference equation holding in each grid point P

$$
\begin{aligned}
&- a_W u_j^i(W) - a_S u_j^i(S) - a_B u_j^i(B) - a_P u_j^i(P) \\
&\quad - a_E u_j^i(E) - a_N u_j^i(N) - a_U u_j^i(U) = b_P, \qquad (25)
\end{aligned}
$$

where

$$
\begin{aligned}
a_W &= \frac{\tau}{h^2}\operatorname{clt}(u_j^{i-1})(P)g(|\nabla u_j^\sigma|)(w), \\
a_E &= \frac{\tau}{h^2}\operatorname{clt}(u_j^{i-1})(P)g(|\nabla u_j^\sigma|)(e), \\
a_S &= \frac{\tau}{h^2}\operatorname{clt}(u_j^{i-1})(P)g(|\nabla u_j^\sigma|)(s), \\
a_N &= \frac{\tau}{h^2}\operatorname{clt}(u_j^{i-1})(P)g(|\nabla u_j^\sigma|)(n), \qquad (26)\\
a_B &= \frac{\tau}{h^2}\operatorname{clt}(u_j^{i-1})(P)g(|\nabla u_j^\sigma|)(b), \\
a_U &= \frac{\tau}{h^2}\operatorname{clt}(u_j^{i-1})(P)g(|\nabla u_j^\sigma|)(u), \\
a_P &= a_W + a_E + a_S + a_N + a_B + a_U + 1, \\
b_p &= u_j^{i-1}(P).
\end{aligned}
$$

Applying the zero Neumann boundary conditions to boundary volumes, we can represent Eq. (25) in matrix form

$$
\mathbf{A}\bar{u}_j^i = \bar{b}, \qquad (27)
$$

where $\bar{u}_j^i$ represents the vector of unknown discrete values of u_j^i in the grid nodes. The coefficients of matrix $\mathbf{A}$ depend on u_j^σ and u_j^{i-1} and thus they are recomputed in each discrete frame and scale step. Because of the dependence on u_j^σ, we have to solve inner Eq. (22). The finite volume method for (22) leads to linear system with the symmetric matrix which is same in each frame and scale step. Hence we obtain the vector $\bar{u}_j^\sigma$ i.e., values of u_j^σ in grid points. Using those values, we compute the approximate value of $g(|\nabla u_j^\sigma|)$ for each crossing finite volume boundary point $w, e, \ldots$ Finally, we arrange matrix $\mathbf{A}$. The system (27) is then solved by efficient linear algebraic solver.

Using similar ideas as in the previous explanation we obtain the discrete version of Eq. (16) as follows

$$
\begin{aligned}
a_W &= \frac{\tau}{h^2}\operatorname{clt}(u_j^{i-1})(P)\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(P)\bigg/\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(w), \\
a_E &= \frac{\tau}{h^2}\operatorname{clt}(u_j^{i-1})(P)\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(P)\bigg/\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(e), \\
a_S &= \frac{\tau}{h^2}\operatorname{clt}(u_j^{i-1})(P)\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(P)\bigg/\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(s), \\
a_N &= \frac{\tau}{h^2}\operatorname{clt}(u_j^{i-1})(P)\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(P)\bigg/\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(n), \\
a_B &= \frac{\tau}{h^2}\operatorname{clt}(u_j^{i-1})(P)\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(P)\bigg/\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(b), \\
a_U &= \frac{\tau}{h^2}\operatorname{clt}(u_j^{i-1})(P)\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(P)\bigg/\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(u), \\
a_P &= \frac{\tau}{h^2}\operatorname{clt}(u_j^{i-1})(P)\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(P)\bigg/\sqrt{\varepsilon^2 + |\nabla u_j^{i-1}|^2}(p), \\
a_P &= a_P + a_E + a_S + a_N + a_B + a_U + 1 \quad b_P = u_j^{i-1}(P).
\end{aligned}
\qquad (28)
$$

Again we end with the linear system of type (27).

Remark. By the construction of system (27) matrix $\mathbf{A}$ is nonsymmetric, but strictly diagonally dominant and thus invertible, so its unique solution always exists. Moreover, the so called $L\,\propto$-stability condition is fulfilled for our discrete solution. Namely, let a, b be real constants. If $a \leqslant \bar{u}_j^0 \leqslant b$, $j = 0, \ldots, m$, then $a \leqslant \bar{u}_j^i \leqslant b\ \forall j = 0, \ldots, m$, $i = 1, 2, \ldots$ [3].

5. Discussion on numerical experiments

In this section we will present and discuss computational results obtained by approximation schemes given in previous sections. First, we are dealing with a phantom-like image sequence consisting of expanding, slightly deforming and moving ellipse with an inner structure in the form of four-petals. We add uniform, impulsive (salt and pepper) and Gaussian noise to frames of the image sequence. The original six-frame sequence and its destroyed version are plotted in the first two columns of Fig. 6. The reconstruction of any noisy frame of the sequence by a standard filtering algorithm (e.g., using commercial software) is a very difficult task and no attempt was successful. The right column of Fig. 6 represents the results of non-linear multi-scale analysis (14) applied to the noisy sequence after 10 discrete scale steps of our algorithm. The similar successful result is obtained using space–time filtering Eq. (16).

Next we have applied multiscale analysis models to an in vivo acquired 3D echo–cardiographic sequence obtained by means of a rotational acquisition technique using the Tom-Tec Imaging System. In Figs. 7–10 (first column), the left ventricular endo-cardium and the left atrium of an entire unfiltered cardiac cycle has been

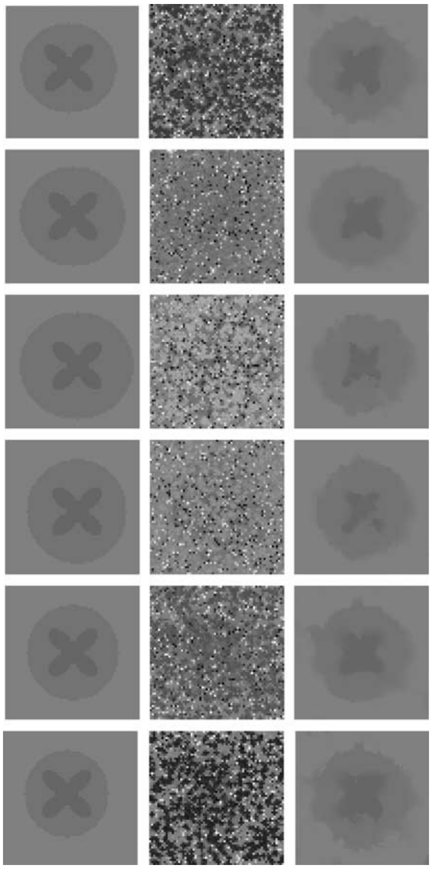

Fig. 6. Two-dimensional phantom: original (left), noisy (middle), and processed (right) images.

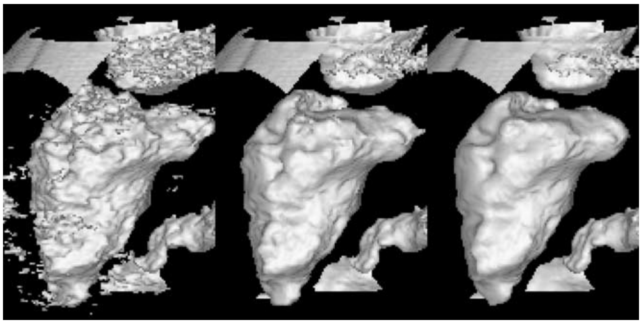

Fig. 7. The multi-scale analysis of the first frame of 3D echo–cardiographic image sequence by Eq. (14).

visualized. The iso-surfaces corresponding to the interface between cardiac muscle and blood have been computed by the marching cubes method and visualized by a Gouraud shading surface rendering. Figs. 7 and 8 consist of three sub-figures. For each row, in the left we plot the echo-volume visualized using the original noisy data, in the middle the result after three discrete scale steps and on the right after nine discrete scale steps of model (14). We have chosen $\tau = 0.2$, $\sigma = 0.0001$, and $h = 1/150$. The next Figs. 9 and 10 are related to the application of model (16). For these computations, we have used parameters $\tau = 0.01$, $\varepsilon = 0.0001$ and $h = 1/150$ and we look at five neighbourhood of each voxel to recognize the Lambertian trajectories of moving points. Again we plot original noisy data (left images) and the results after three (middle images) and six (right images) discrete scale steps of our algorithm for model (16). As one can expect, the resulting shapes are a little bit smoother than in Figs. 7 and 8 due to mean curvature flow effect.

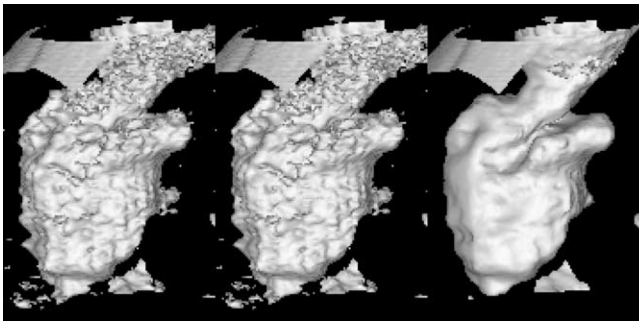

Fig. 8. The multi-scale analysis of the seventh frame of 3D echo–cardiographic image sequence by Eq. (14).

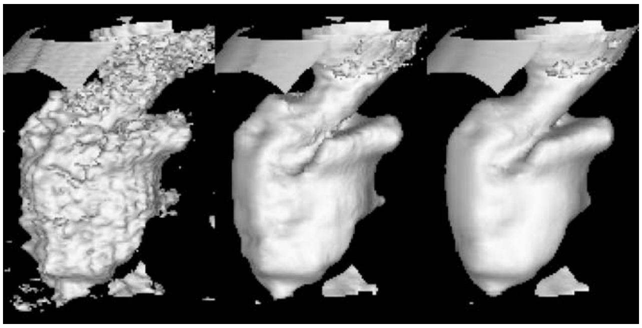

Fig. 9. The multi-scale analysis of the seventh frame of 3D echo–cardiographic image sequence by Eq. (16).

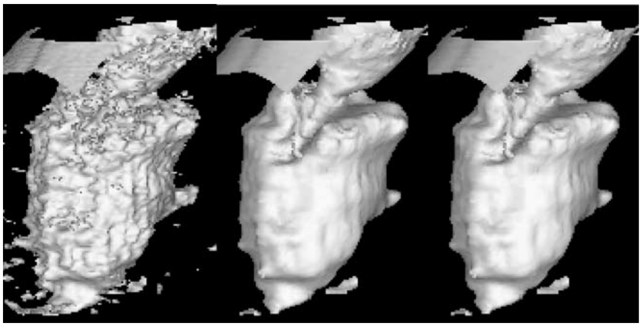

Fig. 10. The multi-scale analysis of the 13th frame of 3D-echo–cardiographic image sequence by Eq. (16).

To evaluate the accuracy of the presented multiscale analysis models, we set the following procedure to define

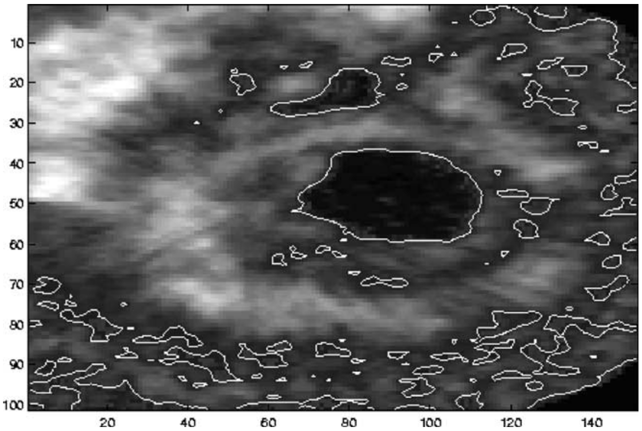

Fig. 11. The level contour line $\tilde{u}$ computed on the unfiltered slice $y = 103$. The gray-level image is visualized in the background.

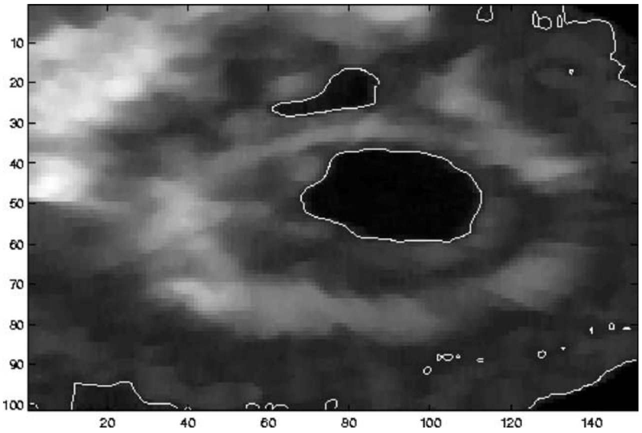

Fig. 12. The level contour line $\tilde{u}$ computed on the slice $y = 103$ after nine discrete scale steps of filtering. The filtered image is shown in the background.

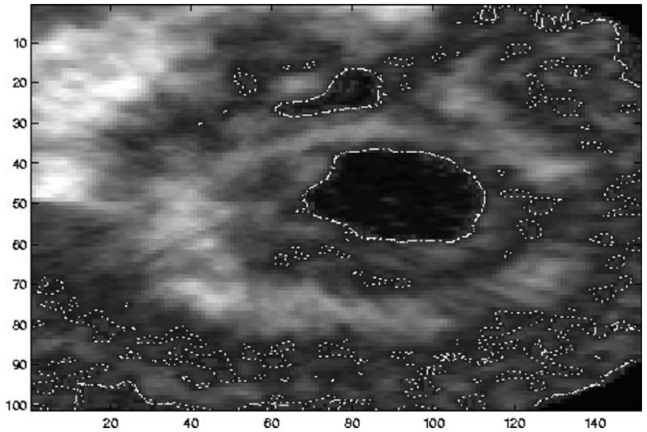

Fig. 13. Comparison between the contour lines corresponding to the unfiltered (dot) and filtered (dot-line) image. In the background, the unfiltered image is shown. For clarity of representation only the subregion around the left and right ventricle is visualized.

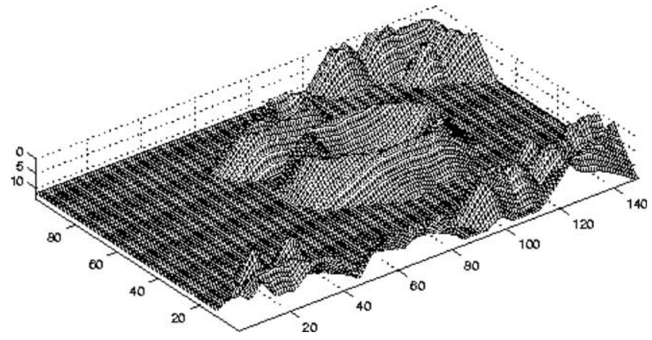

Fig. 14. Distance function $D_l(x, y)$ from the $\tilde{u}$ level set of the filtered image (isoline of Fig. 14). To allow a better visualization the elevation axis has been reversed (0 on the top) and the distance has been cut off for $D_l(x, y) > 10$.

and estimate the error. An external observer has manually segmented the ventricular chamber by considering every single slice l of a 3D unfiltered frame. We compute the mean intensity value $\tilde{u}$ of the points corresponding to the manual segmentation over all the slices and we use $\tilde{u}$ to define the "closest" iso-surface to the manual segmentation itself. In Fig. 11, a slice of a 3D frame with the superimposition of $\tilde{u}$ iso-level curves is shown. In Fig. 12, the same situation after nine discrete scale steps of (14) is reproduced. The non-linear multi-scale analysis does not move significantly the position of the coherent boundaries as is shown in Fig. 13, where the contour lines of the unfiltered and filtered images are compared. At the same time the filtering of the non-coherent structures is performed. Our goal now is to estimate the mean distance between the manual segmentation and the iso-line in both cases before and after filtering. For each slice a distance function $D_l(x, y)$ from the iso-line is computed with level set methods, following [29]. In Fig. 14 the distance function from the iso-line of Fig. 13 is visualized. The mean distance between the manual segmentation curve C_l and the iso-line is defined as $D_l = \int_{C_l} D_l(x, y)\,\mathrm{d}s / \int_{C_l} \mathrm{d}s$, where s is the arclength. In Fig. 15, both C_l and $D_l(x, y)$ are visualized. The distance error has been evaluated for every slice of the 3D dataset. The global distance error is computed by averaging the errors over the L slices, i.e.,

$$D = \frac{1}{L}\sum_{l=1}^{L} D_l.$$

With this procedure we estimated $D = 1.3$ voxels for the unfiltered image and $D = 0.8$ voxels for the result of the multiscale analysis. Such results give a quantitative confirmation of the qualitative visual inspection, about the ability of the method to keep the position of the coherent structures. This quantitative evaluation has to be considered a preliminary result and further quantitative analysis applied to a statistically relevant number of cases is necessary for a full clinical validation.

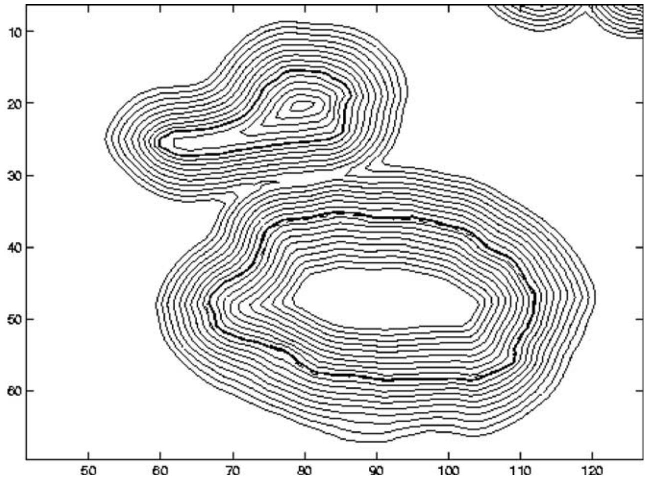

Fig. 15. The manual segmentation curve C_I (dashed line) over-imposed to the distance function $D_I(x, y)$ (solid iso-contour lines). The thick lines represent the zero level set of $D_I(x, y)$.

Recently, the 3D + time filtering method has been applied to real-time 3D echo sequences with quite encouraging results. In Figs. 16–23 we present them for several frames of the entire sequence. In Fig. 16, there are 2D cuts of the subsequent cubes and in Figs. 17–23 we visualize level surfaces corresponding to un-filtered and filtered left ventricular shapes during systole. We have chosen five neighbourhood to compute $\mathrm{clt}(u)$, $\tau = 0.1$, $\sigma = 0.001$, $h = 1/115$, $K = 2$ and the results are plotted after 3 scale steps of the algorithm for Eq. (14).

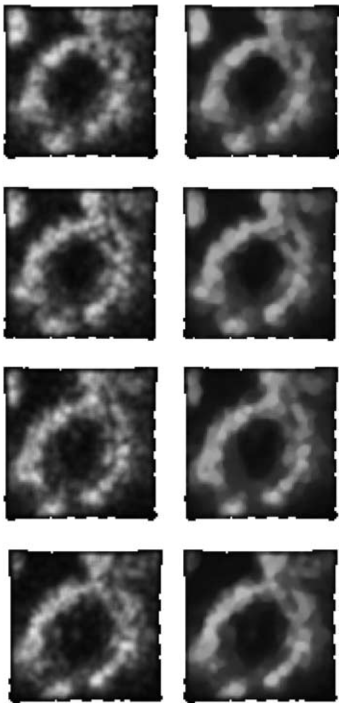

Fig. 16. Processing of RT3DE volumetrics. 2D slices taken from 11th to 14th cubes of the sequence before and after 3D processing by model (14).

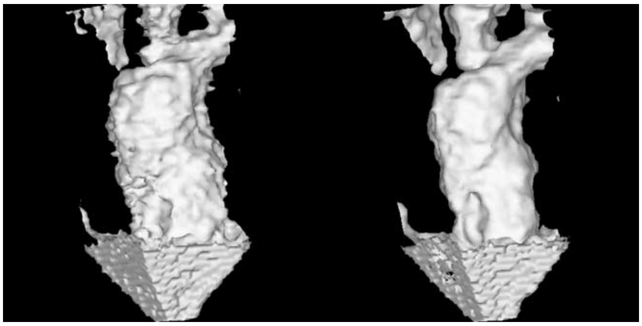

Fig. 17. Processing of RT3DE volumetrics. Visualization of left ventricular surface from fifth cube of the sequence before (left) and after (right) 3D processing by model (14).

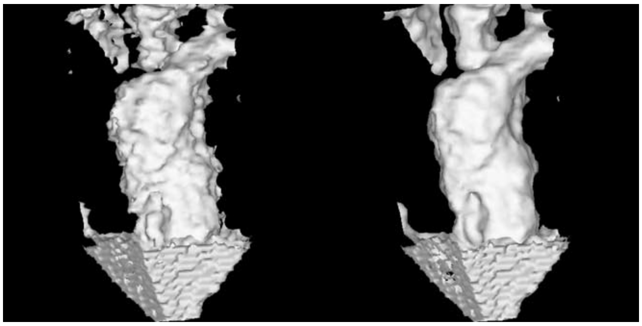

Fig. 18. Processing of RT3DE volumetrics. Visualization of left ventricular surface from sixth cube of the sequence before (left) and after (right) 3D processing by model (14).

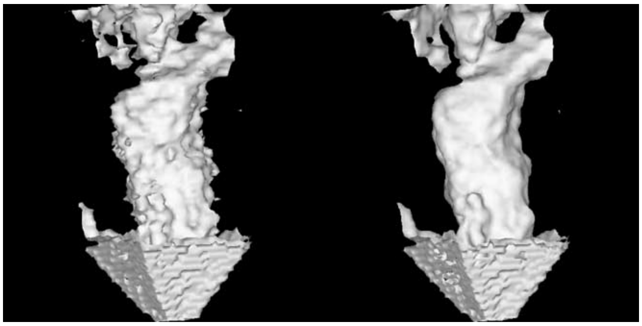

Fig. 19. Processing of RT3DE volumetrics. Visualization of left ventricular surface from seventh cube of the sequence before (left) and after (right) 3D processing by model (14).

6. Preconditioning and solving of linear systems

We will finish with a discussion related to solving linear systems (27) which we shortly indicate as

$$\mathbf{A}u = b \tag{29}$$

by means of iterative techniques, focusing our attention on the use of suitable preconditioners. A preconditioner is a matrix used to accelerate the convergence of the iterative methods, guaranteeing the same solution. The original linear system (29) by the new system

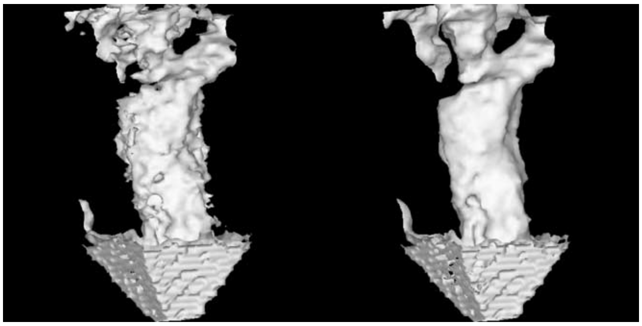

Fig. 20. Processing of RT3DE volumetrics. Visualization of left ventricular surface from eighth cube of the sequence before (left) and after (right) 3D processing by model (14).

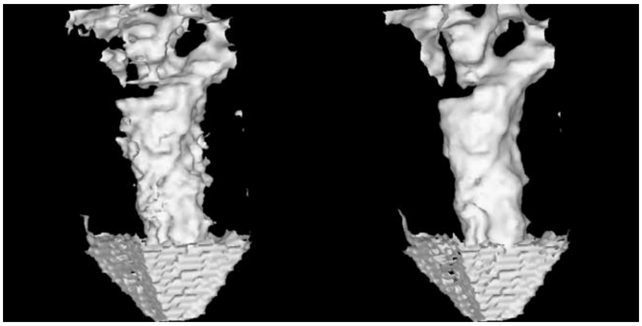

Fig. 21. Processing of RT3DE volumetrics. Visualization of left ventricular surface from nineth cube of the sequence before (left) and after (right) 3D processing by model (14).

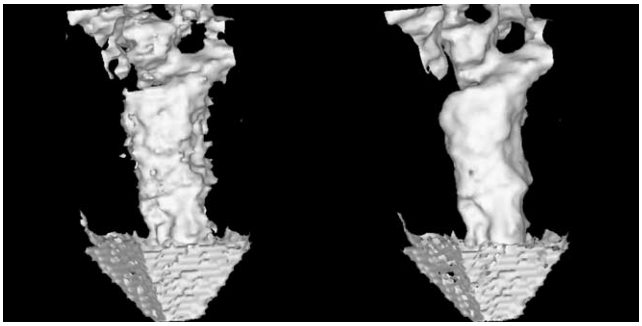

Fig. 22. Processing of RT3DE volumetrics. Visualization of left ventricular surface from 10th cube of the sequence before (left) and after (right) 3D processing by model (14).

$$\mathbf{PAQQ}^{-1}u = \mathbf{P}b, \qquad (30)$$

where matrices **P** and **Q** are the left and right pre-conditioners, respectively. Now we can simply apply an unpreconditioned iterative process to the new system (30). Instead of computing a matrix-vector multiplication $\mathbf{A}v = w$ at each iterative step, now we have to compute also $\mathbf{P}^{-1}w = y$ (or solve the system $\mathbf{P}y = w$) and analogously for **Q**. In devising a pre-conditioner, we are faced by a choice between finding a matrix **P** that approximates $\mathbf{A}^{-1}$, so that only multiplication by **P** is needed, or finding a matrix $\mathbf{P}^{-1}$ that approximates **A**. There are three criteria for choosing an efficient pre-conditioner **P**: the first criterion is that the pre-conditioner **P** must be chosen so that the condition number of $\mathbf{P}^{-1}$ **A** is greatly reduced, thus speeding up the rate of convergence.

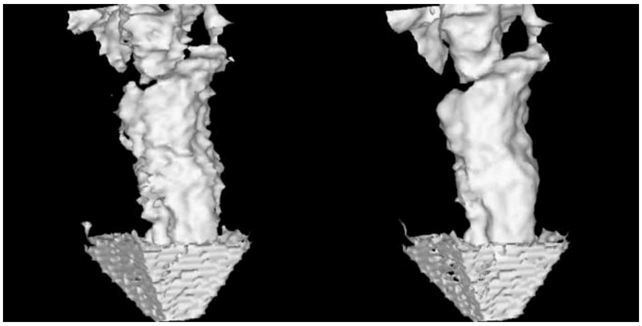

Fig. 23. Processing of RT3DE volumetrics. Visualization of left ventricular surface from 11th cube of the sequence before (left) and after (right) 3D processing by model (14).

The second consideration for a good pre-conditioner is to choose matrix **P** so that the linear system $\mathbf{P}y = w$ can be easily solved. Third, since using a pre-conditioner in an iterative method incurs some extra memory requirement and cost, both initially for the set-up, and per iteration for applying it, the choice for it is a compromise between the cost of constructing and applying the pre-conditioner, and the gain in convergence speed. Certain pre-conditioners need little or no construction phase at all, but for others, such as incomplete factorizations (ILU), there is substantial work involved.

For the 2D image sequences, good performance is achieved also using existing pre-conditioners. In our comparison, we have chosen the classical and most relevant solvers, namely Gauss–Seidel method (GS), conjugate gradient method (CG), biconjugate gradient stabilized method (BICGSTAB), transpose free quasi minimal residual method (TFQMR), generalized minimum residual method (GMRES). At a first investigation we used for each method, when it is possible, the following well-known pre-conditioners: Right ILU(0) (zero fill-in incomplete LU factorization), Left ILU(0), Right MILU(0) (modified ILU(0)), Left MILU(0) and ILUT(n) (ILU with n-fill in and thresholding). For details refer to [51] or [52]. In case of GMRES method, we allow the restart when the dimension of Krylov subspace is 20. We stop the iterations when the Euclidean norm of the actual residual $r^{(k)} = b - \mathbf{A}u_j^{i(k)}$ satisfies

$$\|r^{(k)}\| \leqslant \alpha \|r^{(0)}\|$$

with tolerance parameter $\alpha = 0.01$ and $r^{(0)}$ representing the initial residual. We start with $u_j^{i(0)} = 0$. In the 2D testing example (Table 1), we process a simple phantom given by a moving noisy circle and for report, we choose the time frame from the middle of the sequence. We used $\tau = 0.1$, $\sigma = 0.001$, and $h = 1/128$. In the table, we print

Table 1
Comparison of iterative solvers in one scale step of 2D + time algorithm

No. Multip/CPU time	GS	CG	BICGSTAB	TFQMR	GMRES
No Pre-con	265/25.39	94/6.25	98/7.14	204/17.72	114/19.28
Left ILU(0)		20/2.41	20/2.47	18/2.51	17/3.58
Right ILU(0)		No conver	16/2.03	16/2.32	7/1.13
Left ILU(0)		16/1.95	20/2.48	22/3.05	15/3.02
Right MILU(0)		5/0.69	20/2.46	12/1.72	5/0.8
ILUT(5)		—	10/1.57	8/1.11	3/0.51

the number of multiplications (matrix times vector) performed until convergence is reached because this is the most expensive step of the iterative procedure and this number is independent of computer architecture. We also report the CPU time in seconds (on SUN Sparc 10 workstation) for each complete iterative procedure. This comparison is related also to paper [41] in which an additive operator splitting (AOS) method is used for solving the an-isotropic diffusion equation by a semi-implicit technique. Such a method is direct and thus uses a fixed number of operations. They report that, in general, the AOS scheme is about three times faster than classical Gauss–Seidel method with the tolerance $\alpha = 0.01$. From our comparison, it is clear that any preconditioned iterative method is about 10 times faster than Gauss–Seidel procedure for matrices receiving from image multi-scale analysis models. In the best case, using the GMRES method, we have a speed-up of about 50.

The construction of the next pre-conditioner is based on the concept that for diagonally dominant $\mathbf{A}$ the entries in $\mathbf{A}^{-1}$ decay rapidly away from diagonal and a banded pattern for $\mathbf{P}$ will produce a good approximate inverse.

The rows of the pre-conditioner are generated independently, allowing a parallel computation of them. Let p_i denote the column of $\mathbf{P}^{\mathrm{T}}$. Then ideally we would like to have

$$\mathbf{A}^{\mathrm{T}} p_i = e_i, \tag{31}$$

where e_i is the ith column of the identity matrix. From $\mathbf{A}$, we determine some small list of indices I drawn from $\{1, 2, \ldots, N\}$, where N is equal to the number of unknowns, so that the variables and constraints in I have the most impact on the variable i. Then we solve the small system of equations

$$\overline{\mathbf{A}^{\mathrm{T}}}\,\overline{p_i} = \overline{e_i}, \tag{32}$$

where the bar indicates that we delete all the rows and columns except for those included in I.

Two possibilities are tested for the choice of I for node i. We let I be i together with all indices of neighbours of node i neglecting those, say j, so that

1. $|a_{ij} * a_{ji}| \leqslant thr_1 |a_{ii} * a_{jj}|$,
2. $|a_{ij} + a_{ji}| \leqslant thr_2 |a_{ii} + a_{jj}|$,

where thr_1, thr_2 are user-specified tolerances. The important observation is that the dimension of the small systems does not exceed 7 in 3D computations and so the solution of (32) can be computed in an explicit manner.

After the small system, involving these rows and columns is solved, we scatter the entries of the solution $\bar{p}_i$ back to their original coordinate positions in p_i, and we fill in the remaining positions of p_i with zeros. Using compressed sparse row technique for matrices, we only have to store immediately the computed elements and the I indices. As a final observation we may point out that the small systems to be solved are principal submatrices of $\mathbf{A}$, that is a strictly diagonally dominant M-matrix. So it is easy to verify that the pre-conditioner can be computed in a stable manner.

In the testing of iterative solvers, we stop the iterations by the same criterion as in 2D experimentations. For every method, we print the number of multiplications (matrix times vector) performed until convergence and also CPU time in seconds for each complete iterative procedure. For the tests in 3D, we consider a system arising from the multi-scale analysis of ultrasound image sequence. We consider cubes consisting of 64^3 voxels and we use for the pre-conditioner $thr_1 = 0.05$ (Table 2) and $thr_2 = 0.8$ (Table 3). In that case, we observe a good performance by BICGSTAB and GMRES with left pre-conditioner providing computations on SUN Ultra 30.

Research projects on efficient iterative solvers and pre-conditioners are in progress and particular attention

Table 2
Comparison of iterative solvers with approximate inverse pre-conditioner in the processing of one frame of 3D image sequence using model (14)

No. Multip/CPU time	BICGSTAB	GMRES
No Pre-con	145/152.27	125/274.09
Left	8/8.60	9/14.63
Right	96/76.68	45/101.23

Table 3
Comparison of iterative solvers with approximate inverse pre-conditioner in the processing of one frame of 3D image sequence using model (16)

No. Multip/CPU time	BICGSTAB	GMRES
No Pre-con	145/152.27	125/274.09
Left	7/8.73	7/12.58
Right	121/118.22	64/149.85

is devoted to the solution of very large dimension linear systems, possibly ill-conditioned, by conjugate-gradient methods and the choice of stopping iteration [53–55]. These are crucial tasks to obtain a fast, high quality solution and restored image.

7. Conclusions

The processing of 3D echo–cardiographic sequences, we have proposed uses spatial and temporal information to filter out noise and to preserve the coherent space–time structures. The PDE representing the model combines the effect of the an-isotropic and geometrical diffusion in space and the Galilean invariant movie multi-scale analysis in time. In the limited number of synthesized and real sequences that we have considered the results are encouraging both from the qualitative and quantitative point of view. In particular the boundaries of the coherent structures are not moved significantly while a remarkable reduction of the noise is provided. A robust, efficient and stable numerical scheme to solve the discretized PDE has been proposed and a comparison among several iterative solvers has been performed.

Acknowledgments

The authors would like to thank B. Mumm at TomTec for supplying the sequence of 3D Echo data and J. Thomas at Cleveland Clinic Foundation for supplying real-time 3D echo sequences. The numerical experiments presented in the paper were partly computed using large-scale computational facilities of the Supercomputing group at CINECA, InterUniversity Computing Center, Bologna (Italy). The work has been supported by the Office of Energy Research, Office of Computational and Technology Research, Mathematical, Information, and Computational Science Division, Applied Mathematical Sciences Subprogram, of the US Department of Energy, under Contract No. DE-AC03-76SF00098. The work has also been supported by IPER ICARUS Project and MINOS Project (funded by the European Commission) at CINECA, Bologna, by CNR, Italy, by the Stefan Banach International Mathematical Center as a Center of Excellence (Package 8: Visual Modelling), Warsaw, Poland, and by the grant VEGA 1/7132/20 of the Slovak Scientific Grant Agency, and partially supported by MURST, grant number MM01111258.

References

[1] Mikula K, Sarti A, Lamberti C. Geometrical diffusion in 3D-echocardiography. In: Proceedings of ALGORITMY'97, Conference on Science Compositions, Zuberec, Slovakia. 1997. p. 167–81.

[2] Handlovičová A, Mikula K, Sarti A. Numericalsolution of parabolic equations related to level set formulation of meancurvature flow. Computing and Visualization in Science 1999;1(2):179–82.

[3] Sarti A, Mikula K, Sgallari F. Nonlinear multiscaleanalysis of 3D echocardiographic sequences. IEEE Transactions on Medical Imaging 1999;18(6):453–66.

[4] Sarti A, Ortiz de Solorzano C, Lockett S, Malladi R. A Geometric Model for 3-D Confocal Image Analysis. IEEE Transactions on Biomedical Engineering 2000;45(12):1600–10.

[5] Sarti A, Malladi R, Sethian JA. Subjective surfaces: a method for completing missing boundaries. Proceedings of the National Academy of Sciences of the United States of America 2000;12(97):6258–63.

[6] Sarti A, Malladi R, Sethian JA. Subjective surfaces: a geometric model for boundary completion. International Journal of Computer Vision 2002;46(3):201–21.

[7] A. Sarti, R. Malladi, A geometric level set model for ultrasounds analysis, LBNL-44442, University of California, Berkeley, 1999.

[8] A. Sarti, A. Wiegmann, Edges are image discontinuities-fastedge enhancement based on explicit-jump multiscale analysis, LBNL-42373,University of California, Berkeley, 1999.

[9] Bänsch E, Mikula K. Adaptivity in 3D image processing. Computing and Visualization in Science 2001;4(1):21–30.

[10] Bommer W, Weinert L, Neumann A, Neef J, Mason D, Demaria A. Determination of right atrial and right ventricular size bytwo-dimensional echocardiography. Circulation 1979:60–91.

[11] Baraldi P, Sarti A, Lamberti C, Prandini A, Sgallari F. Evaluation of differential optical flow techniques on synthesized echo images. IEEE Transactions in Biomedical Engineering 1996;43(3).

[12] Maillaux GE, Laglois F, Simard PY, Bertrand M. Restoration of the velocity field of the heart from two-dimensional echocardiograms. IEEE Transanctions in Medical Imaging 1989;8:143–53.

[13] Tomtec Imaging Systems GmbH, ww.tomtec.de..

[14] Smith SW, Pavy Jr. HG, von Ramm OT. High-speed ultrasound-volumetric imaging system part I: transducer design and beam steering. IEEE Transanctions On Ultrasonics, Ferroelectrics and Frequency Control 1991;38:100–8.

[15] von Ramm OT, Smith SW, Pavy Jr. HG. High-speed ultrasound volumetric imaging system part II: parallel processing and image display. IEEE Transanctions On Ultrasonics, Ferroelectrics, and Frequency Control 1991;38:109–15.

[16] Alvarez L, Guichard F, Lions PL, Morel JM. Axioms and fundamental equations of image processing. Archieves of Rational and Mechanical Analysis 1993;123:200–57.

[17] P. Perona, J. Malik, Scale space and edge detection usinganisotropic diffusion, in: Proc. IEEE Computer Society Workshop on Computer Vision, 1987.

[18] Bart ter Haar Romeny M, editor. Geometry Driven Diffusion In Computer Vision. Dordrecht: Kluwer; 1994.

[19] M. Nielsen, P. Johansen, O.F. Olsen, J.Weickert (Eds.), Scale-space Theories in Computer Vision, Lecture Notes in Computer Science, 1682, Springer, Berlin, 1999.

[20] Crandall MG, Ishii H, Lions PL. User's guide to viscosity solutions of second-order partial differential equations. Bulletin (NS) American Mathematics Society 1992;27:1–67.

[21] Handlovičová A, Mikula K, Sgallari F. Variational numerical methods for solving nonlinear diffusion equations arising inimage processing. Journal of Visual Communication and Image Representation 2002;13(1/2):217–37.

[22] Dziuk G. Algorithm for evolutionary surfaces. Numerische Mathematik 1991;58:603–11.

[23] Mikula K, Ševčovič D. Evolution of plane curves driven by a nonlinear function of curvature and anisotropy. SIAM Journal of Applied Mathematics 2001;61(5):1473–501.

[24] Mikula K, Kačur J. Evolution of convex plane curvesdescribing anisotropic motions of phase interfaces. SIAM Journal of Science Composition 1996;17:1302–27.

[25] U. Clarenz, U. Diewald, M. Rumpf, Anisotropic geometric diffusion in surface processing. Preprint University of Bonn, 2000.

[26] Nochetto R, Paolini M, Verdi C. Sharp error analysis forcurvature dependent evolving fronts. Mathematical Models and Methods in Applied Sciences 1993;3(6):711–23.

[27] Osher S, Sethian JA. Front propagating with curvature dependent speed: algorithms based on the Hamilton–Jacobi formulation. Journal of Computer Physics 1988;79:12–49.

[28] Sethian JA. Numerical algorithm for propagating interfaces: Hamilton–Jacobi equations and conservation laws. Journal of Differential Geometry 1990;31:131–61.

[29] Sethian JA. Level Set Methods and Fast Marching Methods. Evolving Interfaces in Computational Geometry, Fluid Mechanics, ComputerVision, and Material Science. Cambridge, UK: Cambridge University Press; 1999.

[30] Malladi R, Sethian JA, Vemuri B. Shape modeling withfront propagation: a level set approach. IEEE Transanctions Pattern Analysis Machine Intelligence 1995;17(2):158–74.

[31] Malladi R, Sethian JA. Image processing: flows under min/max curvature and mean curvature. Graphical Models and Image Processing 1996;38(2):127–41.

[32] Alvarez L, Lions PL, Morel JM. Image selective smoothing and edge detection by nonlinear diffusion. SIAM Journal of Numerical Analysis 1992;29:845–66.

[33] Chen Y-G, Giga Y, Goto S. Uniqueness and existence of viscosity solutions of generalized mean curvature flow equation. Journal of Differential Geomtry 1991;33:749–86.

[34] Evans LC, Spruck J. Motion of level sets by mean curvature. Indian Journal of Differential Geometry 1991;33:635–81.

[35] Catte' F, Lions PL, Morel JM, Coll T. Image selective smoothing and edge detection by nonlinear diffusion. SIAM Journal of Numerical Analysis 1992;29:182–93.

[36] Kačur J, Mikula K. Solution of nonlinear diffusion appearing in image smoothing and edge detection. Applied Numerical Mathematics 1995;17:47–59.

[37] Bänsch E, Mikula K. A coarsening finite element strategy in image selective smoothing. Computing and Visualization in Science 1997;1(1):53–61.

[38] Mikula K, Ramarosy N. Semi-implicit finite volume scheme for solving nonlinear diffusion equations in image processing. Numerische Mathematik 2001;89(3):561–90.

[39] Kriva' Z, Mikula K. An adaptive finite volume scheme for solving non-linear diffusion equations in image processing. Journal of Visual Communication and Image Representation 2002;13(1/2): 22–35.

[40] Kriva' Z, Mikula K. An adaptive finite volume scheme inprocessing of color images. In: Proceedings of ALGORITMY 2000, Conference on Science Composition, Podbanske Slovakia. 2000. p. 174–88.

[41] Weickert J, Romeny BMtH, Viergever MA. Efficient and reliable schemes for nonlinear diffusion filtering. IEEE Transanctions On Image Processing 1998;7.

[42] Preusser T, Rumpf M. An adaptive finite element method for large scale image processing. In: Proceedings of Scale Space'99. 1999. p. 223–34.

[43] Nitzberg M, Shiota T. Nonlinear image filtering with edge and corner enhancement. IEEE Transactions on Pattern Analysis and Machine Intelligence 1992;14(8):826–33.

[44] Weickert J. Anisotropic diffusion in computer vision. Stuttgart: Teubner; 1998.

[45] Caselles V, Morel J-M, Sapiro G, editors. IEEE Transactions on Image Processing, 7. 1998.

[46] Walkington NJ. Algorithms for computing motion by mean curvature. SIAM Journal of Numerical Analysis 1996;33(6):2215–38.

[47] Rudin LI, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms. Physica D 1992;60:259–68.

[48] Alvarez L, Morel JM. Formalization and computational aspects of image analysis. Acta Numerica 1994:1–59.

[49] F. Guichard, Axiomatisation des analyses multi-échellesd'images et de films. PhD Thesis Univerity Paris IX Dauphine (1994).

[50] Patankar S. Numerical heat transfer and fluid flow. Washington, DC: Hemisphere Publishing Company; 1980.

[51] Saad Y. Iterative methods for sparse linear systems. Boston, MA: PWS Publishing Company; 1996.

[52] Golub G, van Loan C. Matrix Computations. third ed. Baltimore, MD: The Johns Hopkins University Press; 1996.

[53] D. Calvetti, S. Morigi, L. Reichel, F. Sgallari, Thikonovregularization and the *L*-curve for large discrete ill-posed problems, Journal of Computer Applications in Mathematics, special volume series: Studies in Computational Mathematics, Numerical Analysis 2000 in the 20th Century, vol. III, Linear Algebra, vol. 123, (1–2), November 2000, pp. 423–446.

[54] D. Calvetti, S. Morigi, L. Reichel, F. Sgallari, An iterative method with error estimators, Journal of Computer Applications in Mathematics, special volumeseries: Studies in Computational Mathematics, Numerical Analysis 2000 in the 20th Century, vol. V, Quadratures and orthogonal polinomials, vol. 127, pp. 93–119.

[55] Calvetti D, Morigi S, Reichel L, Sgallari F. Computable error bounds and estimates for the conjugate gradient method. Numerical Algorithms 2001;25:75–88.