

A new tangentially stabilized 3D curve evolution algorithm and its application in virtual colonoscopy

Karol Mikula · Jozef Urbán

Received: 3 November 2012 / Accepted: 26 September 2013 /
Published online: 22 October 2013
© Springer Science+Business Media New York 2013

Abstract In this paper we develop a new efficient and stable Lagrangian numerical method for computing the evolution of 3D curves driven in the normal plane by a driving force and curvature. This new method contains asymptotically uniform tangential redistribution of grid points designed originally for 3D curve evolution in this paper which makes our computations stable and is crucial for the presented application. Together with the design of new tangentially stabilized algorithm for 3D evolving curves, we develop a new method for the fully automatic finding of the optimal trajectory of the camera in the virtual colonoscopy. The proposed method consists of three steps: 3D segmentation of the colon from CT images, finding an initial trajectory guess inside the segmented 3D subvolumes, and driving the initial 3D curve to its optimal position. To that goal, a suitable intrinsic advection-diffusion partial differential equation with a driving force is designed and solved numerically in fast and robust way in order to find a smooth uniformly discretized 3D curve representing the ideal path of the camera in the virtual colonoscopy.

Keywords Virtual colonoscopy · Evolving 3D curves · Tangential velocity · Asymptotically uniform redistribution · Distance function · Segmentation

Mathematics Subject Classifications (2010) 35R01 · 65M08 · 68U10

Communicated by: K. Urban

K. Mikula (✉) · J. Urbán
Department of Mathematics and Descriptive Geometry, Slovak University of Technology,
Radlinského 11, 813 68 Bratislava, Slovakia
e-mail: karol.mikula@gmail.com
URL: <http://www.math.sk>

J. Urbán
e-mail: jozo.urban@gmail.com

1 Introduction

In this paper we develop a new asymptotically uniform tangential grid point redistribution algorithm for 3D evolving curves in parametric representation. The method is designed for 3D curve evolutions depending on a general velocity field and on curvature. It is based on direct Lagrangean approach for which a suitable tangential redistribution is crucial in order to get fast and stable solution methods. There exist several successful tangential redistributions in 2D for various normal velocities, namely, the ones preserving the ratio of local grid point distances [10, 13, 14], the locally diffusive redistributions [7, 16] and also the redistributions making the local grid point distances uniformly distributed [1, 15, 17]. The asymptotically uniform tangential redistribution was originally introduced in [15] and its description for a very general 2D curve evolution models including driving forces, curvature and the Laplacian of curvature is given in [17]. For the evolution of 3D curves depending on curvature there exists a tangential redistribution method based on a special $\kappa_1 - \kappa_2 - \omega - L$ curve evolution formulation [11] and there is also a special finite element approximation of gradient flows presented in [2].

Our method is designed for 3D curve evolutions driven by the general driving forces given in the form of velocity vector fields and by the curvature. It was inspired by a methodology developed in [11] and it can be understood as a generalization of the asymptotically uniform tangential redistribution method from [15]. Its development was strongly influenced by the application in virtual colonoscopy where our goal was the design of new fast and robust method for the fully automatic extraction of the ideal path of the virtual camera where nor user interaction nor additional parameters are involved. Since we understand the ideal path as the 3D curve which passes along the centerline of the colon, is smooth and uniformly discretized, the smoothly evolved and tangentially redistributed 3D discrete curves are suitable mathematical models to reach that goal. There are three basic steps in our method for finding the ideal path. First, the 3D segmentation of the colon from CT images is performed. Due to the quality of CT data, the classical approaches like the thresholding and the region growing are used, see e.g., [3]. As the result we get all simply-connected parts of the large and small intestines filled with the gas. The next step consists in finding an initial guess for the camera trajectory in every simply-connected segmented subvolume of the intestine. Such initial guess is obtained by using the Dijkstra algorithm [5] for computing an approximate distance from point sources inside the segmented subvolumes followed by the backtracking in the steepest descent direction [21]. The third step is the core of our approach. It consists in driving the initial guess to its optimal position in a smooth and stable way. To that goal we construct a vector field given by the gradient of distance function to the segmented intestine borders which is computed by a 3D generalization of the approach from [4] based on the numerical solution of the time relaxed eikonal equation. Then, our 3D curve evolution algorithm with tangential redistribution is applied to the initial curve. The curve is driven by the velocity given by the projection of the computed distance gradient vector field to the plane normal to the evolving

curve, the evolution is regularized by the curvature, which makes it smooth, and it is accompanied by the suitable choice of the tangential velocity which makes the curve uniformly discretized during the evolution. Such 3D curve evolution model is written in the form of an intrinsic advection-diffusion partial differential equation with a driving force which is then solved by the fast and stable semi-implicit scheme. The resulting smooth uniformly discretized 3D curve representing the ideal trajectory of the camera in virtual colonoscopy is found (performing all steps) in about 8 s on standard PC. Thus the method is highly competitive and is being implemented into the medical software TomoCon of the TatraMed spol s r.o., Bratislava company.

The paper is organized as follows. In Section 2 we briefly comment on the virtual colonoscopy method. In Section 3 we discuss our colon segmentation method. In Section 4 we discuss our method for the fully automatic finding of the optimal camera trajectory. Here we emphasize the design, implementation and results of our new tangentially stabilized 3D curve evolution algorithm presented and discussed in details in Sections 4.4 and 4.5. We also note, that the first presentation of our new method was given in the conference proceedings paper [18].

2 Virtual colonoscopy

A classical optical colonoscopy is an examination of the colon (large intestine) which can successfully detect colon polyps and colorectal tumours. The examination takes 15–60 min and it is performed by a colonoscope which is a flexible tube with a miniature camera and which may also provide a tool for removing a tissue. The colonoscope is introduced into the colon through the rectum, it moves along the colon and a physician can see the situation in the colon on the screen. Because this examination is uncomfortable and painful, the patient receives a medication absorbing the pain or it is done under the general anesthesia. On the other hand, the virtual colonoscopy, introduced in [8], is an examination performed by using the computed tomography (CT). The colon is inflated (with air or CO₂) and then the patient is scanned in two positions (on the abdomen and on the back) by CT. The colon can be viewed similarly to the classical optical colonoscopy, but the physician controls the so-called virtual camera by using its computed "ideal" trajectory. The computation of the ideal path is the important part of the process and an illustration of approach described in this paper is given in Fig. 1. We can see visualization of the colon border and in green the extracted optimal camera trajectory together with the red arrow indicating a polyp found by the voyage along the virtual path. By the literature, the results achieved by the virtual colonoscopy are comparable to the classical approach [12]. Moreover, the images can be viewed at any time, it provides the option to view panoramas of the colon surface, to make its unfolding etc. In addition to these benefits, the virtual colonoscopy allows to examine the colon parts impassable for the colonoscope and it avoids a risk of perforation of the colon. A disadvantage is the radiation during CT examination and the fact that if the physician has found a polyp or tumour, it cannot be removed by the virtual approach.

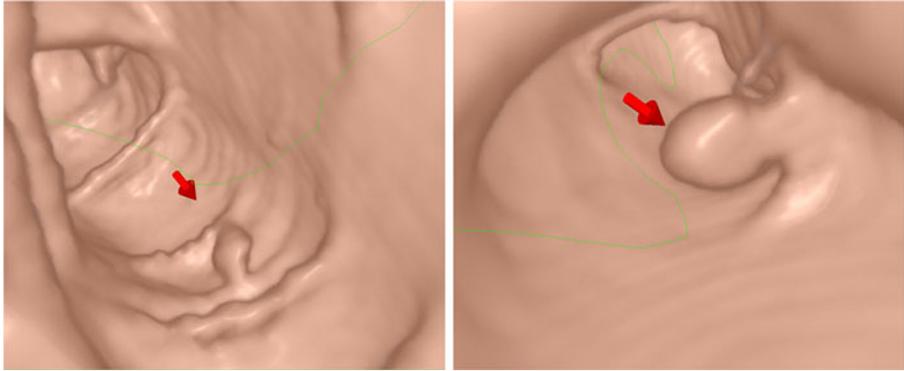


Fig. 1 Polyps found in the large intestine using the virtual colonoscopy

3 The colon segmentation and the initial trajectory guess

The 3D colon image data sets obtained by CT are given by a sequence of 2D slices (512×512 pixels) with a typical slice thickness about 0.75 mm. For our further goals, it is sufficient to subsample data and work with 3D images of the typical size $256 \times 256 \times 400$ voxels. First, we use a thresholding corresponding to the air (about -1000 HU) in CT scans and detect all subvolumes filled by the gas. All voxels of these subvolumes get value 1, the others were set to 0. Next, we apply the region growing method in order to find all simply connected parts of the large and small intestines. The first seed is put to a corner of the 3D image and the region growing algorithm finds all voxels outside the body, their value is put to 0, so this subvolume is ignored. Next, we go subsequently through the whole 3D image and the seed for the next region growing is the next voxel found with the value 1. This seed and all voxels found by the region growing get number 2 which is set also as the number of this first inner body subvolume. We continue such procedure until all seeds for the next region growings are found. During the current region growing all detected voxel values are set to the number of the currently segmented subvolume which is given by the increment of the previously detected subvolume number. We also count the number of detected voxels in each subvolume which gives us the approximate size of the segmented structures. The last segmentation step consists in removing all spuriously detected subvolumes inside the body. By checking of the size, we remove small inner structures filled by the gas (detected e.g., in lungs). Then we compute the distance function of all inner voxels to the border of the segmented subvolume (by the method from Section 4.1) and if the global maximum of the distance function (maximal thickness of the structure) is less than a prescribed threshold we ignore such subvolume (representing e.g., the gas between the body and the CT desk). In such way we end up with one (rarely) or several simply connected subregions of the colon (and also of the small intestine) for which we find then the optimal virtual camera trajectories. The visualization of our segmentation results is presented in Fig. 2. The segmented subvolumes are parts of the small and large intestine inflated with the air.

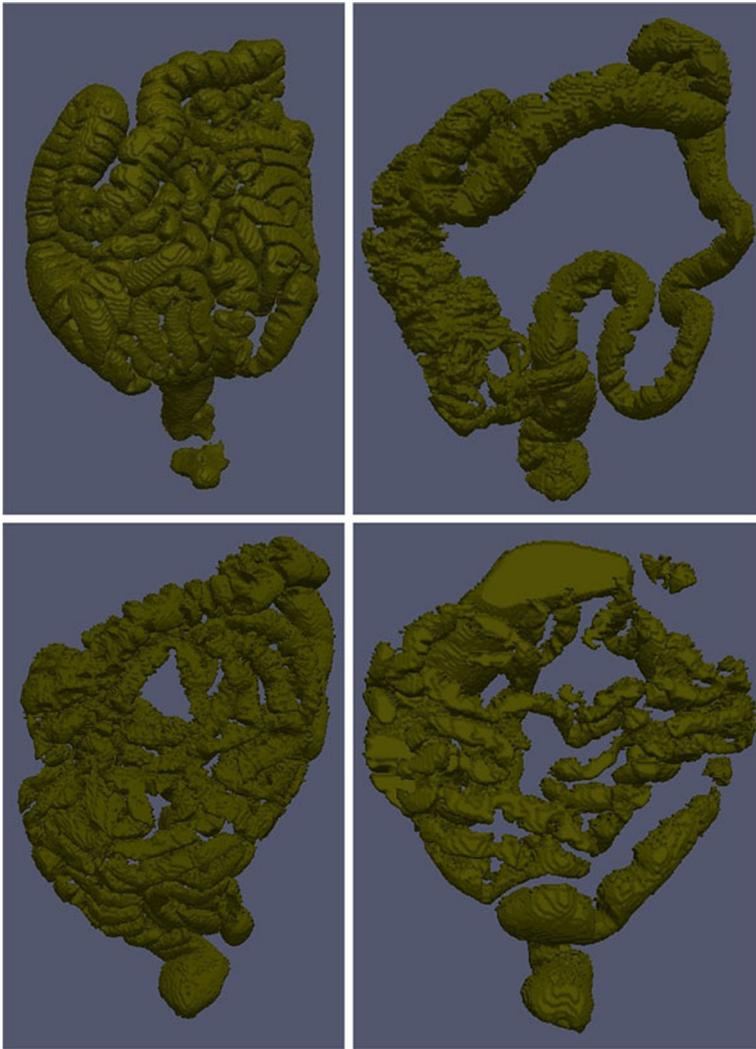


Fig. 2 The examples of segmentation of the large and small intestines

The initial trajectory guess in any colon subvolume is constructed by computing a distance from a point source by the Dijkstra algorithm (in which the graph edges connecting neighbouring voxels have value 1) followed by the backtracking. We note that instead of Dijkstra's algorithm the fast marching method [20] can be also used.

First, we take any point of the subvolume and fix the distance in this point to 0. Then we compute the distance, in the sense of the Dijkstra algorithm, to this fixed point for all voxels inside the subvolume. Since the colon is ablong organ, we take the point with the maximal distance as the first endpoint of the segmented subvolume. Now, we fix the zero value at this first endpoint and use the Dijkstra algorithm again, the point with the maximal distance from the first endpoint will represent the

second endpoint of the subvolume. From this second endpoint we start the backtracking of the computed distances in the steepest descent direction, we end up in the first endpoint of the subvolume. The voxel coordinates of such descending path represent the parametrized 3D curve, the initial guess of the virtual camera trajectory inside the subvolume.

It is clear that the approach described in the above paragraph is fully automatic. For every subvolume of the colon we get the parametrized 3D curve localized inside and connecting its two endpoints. It is important to note that our method is robust with respect to a position and discretization of such initial trajectory guess because the final trajectory will be made smooth, uniformly discretized and centered inside the colon subvolume by a suitable 3D curve evolution model designed in the next section. The only important point is that we have obtained the parametric 3D curve which thus can be used as the initial condition for the suggested 3D curve evolution model.

In order to illustrate and test the particular steps of our method, we constructed 2D and 3D artificial data shown in Fig. 3, the connected circles on the left mimic the typically alternating very thin and thick colon parts (mostly problematic for the algorithms) and U-like volume on the right mimics an overall colon shape. As one can see in Figs. 4–5, the initial trajectory guess is nor smooth nor centered, it touches very often the boundary of the segmented volume. On the other hand, it gives the first parametric representation of the 3D curve which can be evolved to the optimal position by the approach discussed in the next section.

4 Finding the optimal camera trajectory

In this section we discuss important issues leading to a suitable 3D curve evolution model which will drive the initial curve to its optimal position. Our model will be based on a careful construction of the velocity in normal direction, on the regularization of the motion by curvature and on the suitable tangential velocity yielding the

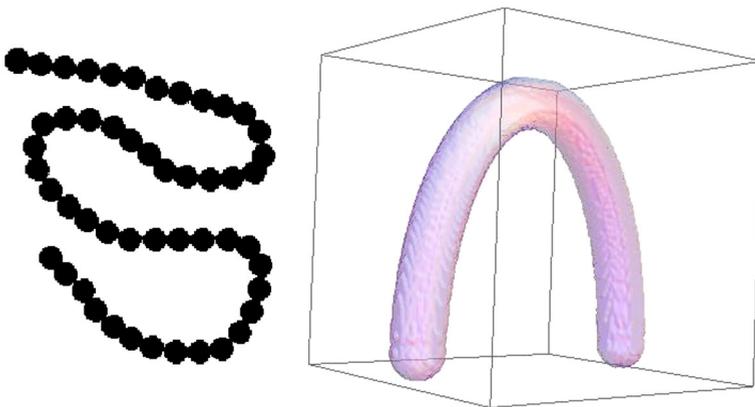


Fig. 3 The 2D and 3D image data sets used for testing the proposed method

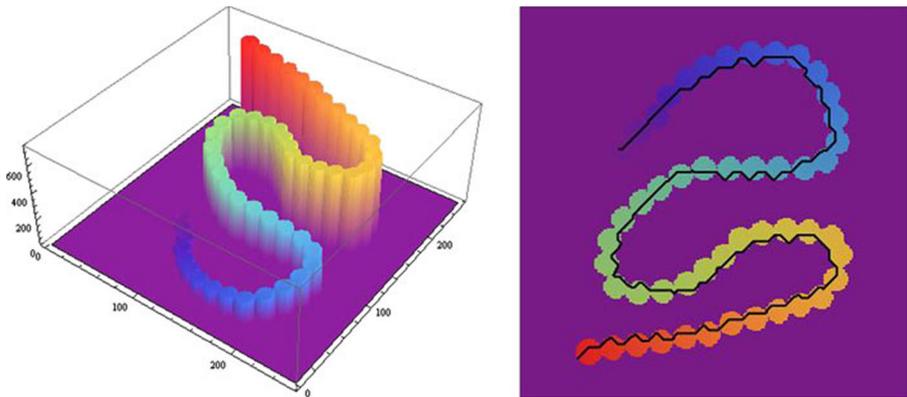


Fig. 4 The graph of the distances (left) and the initial trajectory guess (right)

uniform discretization of the evolving curve. We show that all these issues are necessary ingredients in order to get the smooth and correctly centered virtual camera trajectory in the real colon data of complicated shape.

4.1 Construction of the velocity vector field

In order to get the velocity field by which the 3D curve will be moving to its optimal position we solve the eikonal equation with the zero fixed values in the boundary voxels of the segmented subvolume. Its solution is a distance function which has a ridge along the centerline of the segmented subvolume and the gradient of such distance function points towards the ridge. The initial 3D curve should be driven in a smooth way into that ridge position. Our method for finding the distance function,

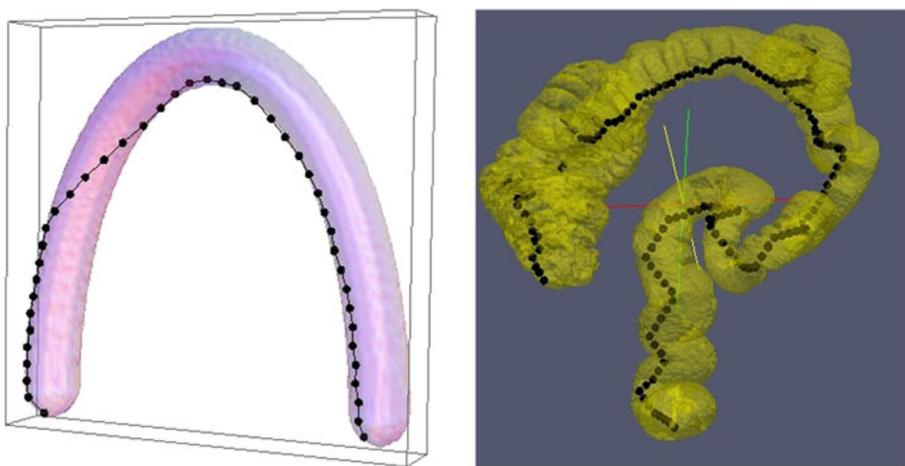


Fig. 5 Initial 3D curve in the test data (left) and in the real segmented colon (right)

which gives the above mentioned vector field, is based on the numerical solution of the time relaxed eikonal equation

$$d_t + |\nabla d| = 1 \tag{1}$$

by the so-called Roy-Tourin scheme [4, 19]. Let us denote by d_{ijk}^n approximate solution d in time step n in the middle of voxel with spatial coordinates (i, j, k) , τ_D the length of the time step and h_D the size of the voxel. Let us define expressions M_{ijk}^{pqr} , where $p, q, r \in \{-1, 0, 1\}$, $|p| + |q| + |r| = 1$, by

$$M_{ijk}^{pqr} = \left(\min \left(d_{i+p, j+q, k+r}^n - d_{ijk}^n, 0 \right) \right)^2. \tag{2}$$

Then the scheme for solving (1) is given by

$$d_{ijk}^{n+1} = d_{ijk}^n + \tau_D - \frac{\tau_D}{h_D} \sqrt{\max \left(M_{ijk}^{-1,0,0}, M_{ijk}^{1,0,0} \right) + \max \left(M_{ijk}^{0,-1,0}, M_{ijk}^{0,1,0} \right) + \max \left(M_{ijk}^{0,0,-1}, M_{ijk}^{0,0,1} \right)}. \tag{3}$$

The values at specified points to which the distance is computed numerically are fixed to zero. In all other points the numerical values are increasing monotonically and if they become changeless we can fix them on the fly [4]. Since the colon is an ablong organ, the method (3) is sufficiently fast and easily implementable and applicable to any complicated shape. After computing the distance function we compute the vector field $\mathbf{v} = \nabla d$ by using the central finite difference approximation of the partial derivatives. In Figs. 6–7 we show visualization of the computed 2D distance function and the associated vector field.

4.2 3D curve evolution in a vector field

In the parametric Lagrangian approach presented in this paper, the evolving 3D curve is represented by discrete points $\mathbf{r}_i^n = (x_i^n, y_i^n, z_i^n)$, where $i = 0, \dots, m$, denotes the grid point number and n represents the discrete time stepping. We consider that the

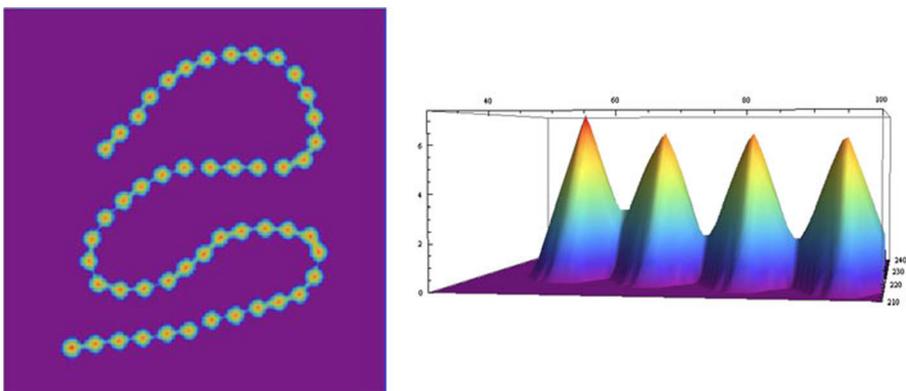


Fig. 6 The distance function to the boundary of 2D testing shape (left) and its detailed graph (right)

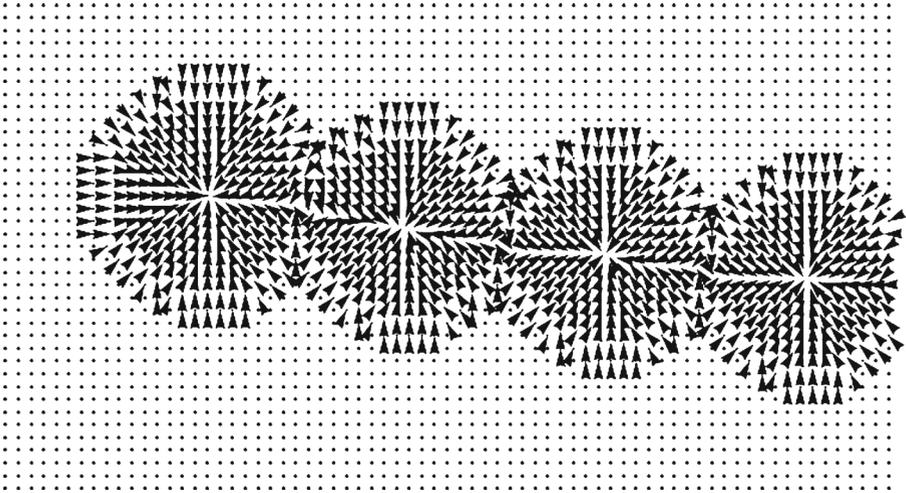


Fig. 7 The detail of the vector field given by the gradient of the distance function

endpoints of the curve (i.e. the 0-th and the m -th point) are fixed. The simplest model for the motion of the curve in the vector field \mathbf{v} is given by

$$\partial_t \mathbf{r} = \mathbf{v}(\mathbf{r}), \quad (4)$$

the numerical discretization of which can be written as

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \tau \mathbf{v}(\mathbf{r}_i^n), \quad (5)$$

where τ being a discrete time step. The results achieved by this approach can be seen in Fig. 8 where all the grid points were moved into the ridge position, but due to the specific direction and length of the velocity field (which is nonzero also on

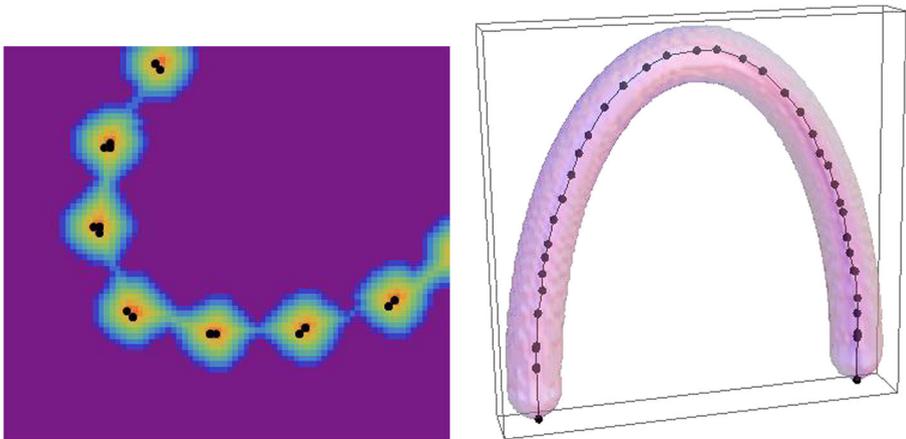


Fig. 8 The results obtained using the velocity field given by the gradient of the distance to the boundary of segmented object in 2D (left) and 3D (right) test data

the ridge), they are packed together and thus it is difficult to get the smooth virtual camera path by using such final curve state. A group of points may contain curve self-intersections (due to numerics) and their distances are irregular so there is no guarantee that the curve would not cross the edge of the colon on the way between the far-distant points. Such situation is not rare in the real data where the colon has complicated structure similar to our connected circles testing example. We note that many standard approaches to virtual colonoscopy, see e.g., [9, 22], uses a combination (e.g., a weighted sum) of two distance functions, the one constructed in the previous section (distance to one fixed endpoint) and the one computed here (distance to the colon borders), followed by a minimization procedure. As we can see, it can lead to a serious troubles, in trajectory representation or to stacking in a local minimum, which are then solved by some heuristic and/or semi-automatic approaches.

The main difficulty of the above simple approach is given by the fact that the grid points just moved independently on each other by the numerical discretization of ODE in direction of the basic velocity field \mathbf{v} . There is no mechanism by which the neighbouring points influence each other and thus move smoothly without degeneracy of their distances. All these problems will be solved, without any heuristic, by our new approach described below.

4.3 Adding the curvature regularization

We know that the motion of the curve can be decomposed into the movement in tangential and normal directions and that the overall shape of the evolving curve with the fixed endpoints is determined by the normal component of the velocity only. The tangential velocity influences the redistribution of points along the curve, thus if it is not controlled, it can cause the accumulation of grid points as in the above mentioned examples. As the first modification of the vector field \mathbf{v} we shall consider its projection to the evolving curve normal plane and thus removing completely the unsuitable tangential motion. This makes the model nonlinear (because the curve normal plane depends on the current curve shape) but it greatly improves the result. Moreover, if we want that the evolving curve points are tied together we have to move from the ordinary to a partial differential equation. A natural intrinsic PDE arising in this case is the one obtained by adding the curvature regularization to the motion by using the curvature vector $k\mathbf{N}$ which is again in the curve normal plane. Let \mathbf{T} be the unit tangent vector to the curve, the projection of vector field \mathbf{v} to the curve normal plane is then defined by

$$\mathbf{N}_v = \mathbf{v} - (\mathbf{T} \cdot \mathbf{v})\mathbf{T}, \tag{6}$$

see Fig. 9, and the regularized motion of the curve in the normal plane is given by

$$\partial_t \mathbf{r} = \mu \mathbf{N}_v + \epsilon k \mathbf{N}, \tag{7}$$

where μ and ϵ are the model parameters. We can consider its explicit numerical discretization in the form

$$\frac{\mathbf{r}_i^{n+1} - \mathbf{r}_i^n}{\tau} = \mu (\mathbf{N}_v)_i^n + \epsilon \frac{2}{h_{i+1}^n + h_i^n} \left(\frac{\mathbf{r}_{i+1}^n - \mathbf{r}_i^n}{h_{i+1}^n} - \frac{\mathbf{r}_i^n - \mathbf{r}_{i-1}^n}{h_i^n} \right), \tag{8}$$

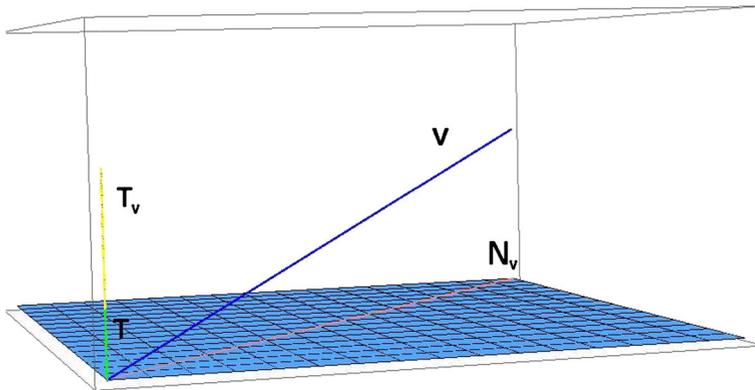


Fig. 9 Visualization of the curve normal plane

for $i = 1, \dots, m - 1$, where the second term on the right hand side represents the discretization of the curvature vector $k\mathbf{N}$, see e.g., [6], and the first term is the approximation of the vector \mathbf{N}_v at the i -th curve grid point, both at the previous time step n . The distances between the grid points are given by the expressions

$$h_i^n = \sqrt{(x_i^n - x_{i-1}^n)^2 + (y_i^n - y_{i-1}^n)^2 + (z_i^n - z_{i-1}^n)^2}. \quad (9)$$

Since we removed the unsuitable tangential component of the velocity and used the curvature regularization, the ridge in the testing data is found in much more regular way, see Fig. 10. The only problem which is still remaining is the nonuniform distribution of the grid points at the final state and also during the subsequent curve evolution which may cause problems during the motion inside complicated shapes. The uniform curve representation would guarantee that the properties of the projected vector field are taking into account uniformly and thus the motion of the curve is

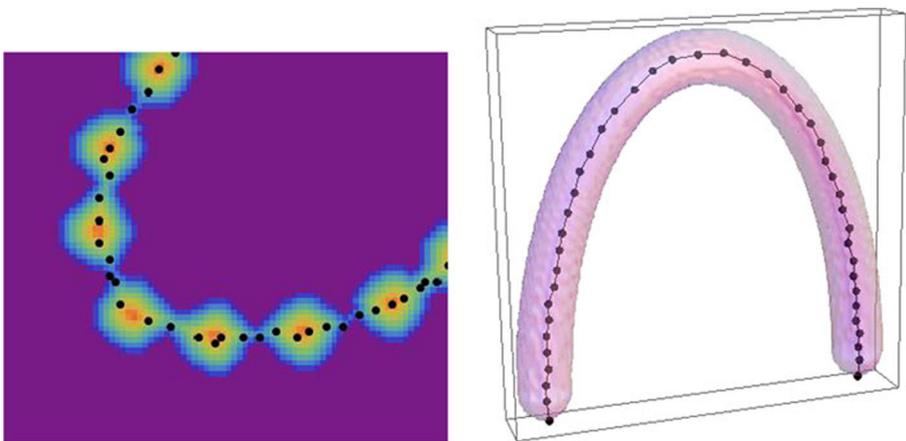


Fig. 10 The results obtained using the projection of the original vector field into the normal plane to the evolving 3D curve accompanied by the curvature regularization

done in the most correct way. The uniform curve discretization will be controlled by adding a suitable tangential velocity into the mathematical model and its numerical discretization.

4.4 Asymptotically uniform tangential redistribution of grid points

Let us consider Eq. 7 accompanied by a controlled tangential motion in the form

$$\partial_t \mathbf{r} = \mu \mathbf{N}_v + \epsilon k \mathbf{N} + \alpha \mathbf{T}. \tag{10}$$

In order to determine a suitable tangential velocity α we introduce the local orthogonal basis smoothly varying along the 3D curve, cf. [11]. In our approach, it will consist of tangent vector \mathbf{T} and two orthogonal vectors in the normal plane defined as

$$\mathbf{N}_1 = \frac{\mathbf{N}_v}{|\mathbf{N}_v|}, \quad \mathbf{N}_2 = \mathbf{N}_1 \times \mathbf{T}. \tag{11}$$

If $|\mathbf{N}_v| = 0$ we redefine \mathbf{N}_1 due to the smoothness requirement, e.g., in discrete settings by the averaged value from the neighboring grid points. Let us define

$$k_1 = k \mathbf{N} \cdot \mathbf{N}_1, \quad k_2 = k \mathbf{N} \cdot \mathbf{N}_2, \tag{12}$$

the projections of the curvature vector onto \mathbf{N}_1 and \mathbf{N}_2 . Then the curvature vector satisfies $k \mathbf{N} = k_1 \mathbf{N}_1 + k_2 \mathbf{N}_2$ and the evolution Eq. 10 can be written as

$$\partial_t \mathbf{r} = U \mathbf{N}_1 + V \mathbf{N}_2 + \alpha \mathbf{T}, \tag{13}$$

with the free parameter α representing the tangential component of the velocity, and with the normal components given by

$$U = \epsilon k_1 + \mu |\mathbf{N}_v|, \quad V = \epsilon k_2. \tag{14}$$

Let us consider the curve Γ with the fixed endpoints parametrized by the position vector \mathbf{r} , and define its local length

$$g = |\mathbf{r}_u| = \left| \frac{\partial \mathbf{r}}{\partial u} \right| = \frac{\partial s}{\partial u}, \tag{15}$$

where $u \in [0, 1]$ and s is the arclength parameter. In discrete settings it is approximated by $g \approx \frac{|\mathbf{r}_i - \mathbf{r}_{i-1}|}{h}$ with $h = \frac{1}{m}$ and this quantity is useful in design of the asymptotically uniform tangential grid point redistribution. We will study the time evolution of the local length g . To that goal we compute

$$g_t = |\mathbf{r}_u|_t = \frac{\mathbf{r}_u}{|\mathbf{r}_u|} \cdot (\mathbf{r}_u)_t \tag{16}$$

and we also have

$$\mathbf{r}_u = \frac{\partial \mathbf{r}}{\partial s} \frac{\partial s}{\partial u} = g \mathbf{T}, \tag{17}$$

$$(\mathbf{r}_u)_t = (\mathbf{r}_t)_u = (U \mathbf{N}_1 + V \mathbf{N}_2 + \alpha \mathbf{T})_u = g (U \mathbf{N}_1 + V \mathbf{N}_2 + \alpha \mathbf{T})_s. \tag{18}$$

Then

$$\begin{aligned}
 g_t &= g\mathbf{T} \cdot (U\mathbf{N}_1 + V\mathbf{N}_2 + \alpha\mathbf{T})_s \\
 &= g\mathbf{T} \cdot (U_s\mathbf{N}_1 + U(\mathbf{N}_1)_s + V_s\mathbf{N}_2 + V(\mathbf{N}_2)_s + \alpha_s\mathbf{T} + \alpha\mathbf{T}_s) \\
 &= gU\mathbf{T} \cdot (\mathbf{N}_1)_s + gV\mathbf{T} \cdot (\mathbf{N}_2)_s + g\alpha_s\mathbf{T} \cdot \mathbf{T} + g\alpha\mathbf{T} \cdot \mathbf{T}_s.
 \end{aligned}
 \tag{19}$$

Since $\mathbf{T} \cdot \mathbf{T} = 1$ and thus $(\mathbf{T} \cdot \mathbf{T})_s = 0$ we have $\mathbf{T}_s \cdot \mathbf{T} + \mathbf{T} \cdot \mathbf{T}_s = 0$ from where $\mathbf{T} \cdot \mathbf{T}_s = 0$. Further, by the Frenet formula we have $\mathbf{T}_s = k\mathbf{N} = k_1\mathbf{N}_1 + k_2\mathbf{N}_2$ and since $\mathbf{N}_1 \cdot \mathbf{T} = 0$ we have

$$\mathbf{T} \cdot (\mathbf{N}_1)_s = -\mathbf{N}_1 \cdot \mathbf{T}_s = -\mathbf{N}_1 \cdot (k_1\mathbf{N}_1 + k_2\mathbf{N}_2) = -k_1$$

and similarly $\mathbf{T} \cdot (\mathbf{N}_2)_s = -k_2$. So finally we get from Eq. 19 the equation for the local length time evolution

$$\partial_t g = g\partial_s \alpha - g(Uk_1 + Vk_2).
 \tag{20}$$

Integrating this equation along the curve i.e. for $u \in [0, 1]$ and considering the fixed endpoints with $\alpha(0) = 0$ and $\alpha(1) = 0$, we get also the time evolution for the total length L

$$\frac{dL}{dt} = - \langle Uk_1 + Vk_2 \rangle_\Gamma L
 \tag{21}$$

where

$$\langle Uk_1 + Vk_2 \rangle_\Gamma = \frac{1}{L} \int_\Gamma Uk_1 + Vk_2 ds$$

denotes the quantity $Uk_1 + Vk_2$ averaged along the curve Γ . In order to define the tangential velocity leading to the asymptotically uniform grid points redistribution, it is worth to study the fraction

$$\frac{g}{L} \approx \frac{|\mathbf{r}_i - \mathbf{r}_{i-1}|}{Lh} = \frac{|\mathbf{r}_i - \mathbf{r}_{i-1}|}{\left(\frac{L}{m}\right)} = \frac{h_i}{\left(\frac{L}{m}\right)}
 \tag{22}$$

representing in discrete settings the ratio of the actual and averaged length of the curve segments. The goal is to design a model in which this ratio tends to 1 or such that the quantity $\theta = \ln\left(\frac{g}{L}\right)$ converges to 0. Using Eqs. 20 and 21 we get for its time evolution the relation

$$\partial_t \theta = \partial_s \alpha - (Uk_1 + Vk_2) + \langle Uk_1 + Vk_2 \rangle_\Gamma.
 \tag{23}$$

On the other hand, if we set $\partial_t \theta = (e^{-\theta} - 1)\omega_r$, where ω_r is a speed of redistribution process, we get that $\theta \rightarrow 0$ as $t \rightarrow \infty$ and we obtain equation for the tangential velocity α guaranteeing the asymptotically uniform redistribution of 3D curve grid points

$$\partial_s \alpha = Uk_1 + Vk_2 - \langle Uk_1 + Vk_2 \rangle_\Gamma + \left(\frac{L}{g} - 1\right)\omega_r.
 \tag{24}$$

4.5 The intrinsic PDE and its numerical discretization

As our final 3D curve evolution model we consider the Eq. 10 with the tangential velocity discussed above. Since $\mathbf{T} = \partial_s \mathbf{r}$ and $k\mathbf{N} = \partial_{ss} \mathbf{r}$ we get the final model in the form of the following intrinsic advection-diffusion PDE with a driving force

$$\partial_t \mathbf{r} = \mu \mathbf{N}_v + \epsilon \partial_{ss} \mathbf{r} + \alpha \partial_s \mathbf{r} \tag{25}$$

with α given by Eq. 24 and accompanied by the Dirichlet boundary conditions (fixed endpoints of the curve). Our final step is the numerical discretization of Eqs. 24–25 by the so-called flowing finite volume method [14, 17] which is adopted here to the 3D curve evolution and, due to a good stability and efficiency properties, we use the so-called semi-implicit approach. Details of such discretization are presented in the sequel.

Let us consider a time difference $\frac{\mathbf{r}^{n+1} - \mathbf{r}^n}{\tau}$ instead of the time derivative in Eq. 25, where τ denotes a uniform discrete time step. Let us denote by $\left[\mathbf{r}_{i-\frac{1}{2}}^n, \mathbf{r}_{i+\frac{1}{2}}^n \right]$ a (dual) segment of the curve Γ at the time step $t_n = n\tau$, where $\mathbf{r}_{i-\frac{1}{2}}^n$ is the middle point of the (primal) segment $\left[\mathbf{r}_{i-1}^n, \mathbf{r}_i^n \right]$. Discretization of the intrinsic PDE (25) is obtained by an integration over the dual segment and taking linear terms at the current time step while the nonlinear terms at the previous time step. First we get

$$\int_{\mathbf{r}_{i-\frac{1}{2}}^n}^{\mathbf{r}_{i+\frac{1}{2}}^n} \frac{\mathbf{r}^{n+1} - \mathbf{r}^n}{\tau} ds = \int_{\mathbf{r}_{i-\frac{1}{2}}^n}^{\mathbf{r}_{i+\frac{1}{2}}^n} \mu (\mathbf{N}_v)^n ds + \int_{\mathbf{r}_{i-\frac{1}{2}}^n}^{\mathbf{r}_{i+\frac{1}{2}}^n} \epsilon \partial_{ss} \mathbf{r}^{n+1} ds + \int_{\mathbf{r}_{i-\frac{1}{2}}^n}^{\mathbf{r}_{i+\frac{1}{2}}^n} \alpha^n \partial_s \mathbf{r}^{n+1} ds \tag{26}$$

where by $\int_{\mathbf{r}_{i-\frac{1}{2}}^n}^{\mathbf{r}_{i+\frac{1}{2}}^n} \psi ds$ we denote an integral of the quantity ψ over the corresponding curve arc. Since the length of the dual segment can be approximated by

$$\left| \mathbf{r}_{i+\frac{1}{2}}^n - \mathbf{r}_{i-\frac{1}{2}}^n \right| = \frac{h_i^n + h_{i+1}^n}{2} \tag{27}$$

where

$$h_i^n = |\mathbf{r}_i^n - \mathbf{r}_{i-1}^n| = \sqrt{(x_i^n - x_{i-1}^n)^2 + (y_i^n - y_{i-1}^n)^2 + (z_i^n - z_{i-1}^n)^2} \tag{28}$$

are Euclidean distances between the grid points approximating lengths of the curve arcs, we get from Eq. 26

$$\frac{h_{i+1}^n + h_i^n}{2} \frac{\mathbf{r}_i^{n+1} - \mathbf{r}_i^n}{\tau} = \mu \frac{h_{i+1}^n + h_i^n}{2} (\mathbf{N}_v)_i^n + \epsilon \left[\partial_{ss} \mathbf{r}^{n+1} \right]_{\mathbf{r}_{i-\frac{1}{2}}^n}^{\mathbf{r}_{i+\frac{1}{2}}^n} + \alpha_i^n \left(\mathbf{r}_{i+\frac{1}{2}}^n - \mathbf{r}_{i-\frac{1}{2}}^n \right)$$

where $(\mathbf{N}_v)_i^n$ and α_i^n represent the normal velocity vector \mathbf{N}_v and the tangential velocity α at the grid point \mathbf{r}_i^n . Finally, when we approximate the first derivatives in the

second term on the right hand side by the finite differences and consider the average of grid points in the third term we get the system of equations

$$\frac{h_{i+1}^n + h_i^n}{2} \frac{\mathbf{r}_i^{n+1} - \mathbf{r}_i^n}{\tau} = \mu \frac{h_{i+1}^n + h_i^n}{2} (\mathbf{N}_v)_i^n + \epsilon \left(\frac{\mathbf{r}_{i+1}^{n+1} - \mathbf{r}_i^{n+1}}{h_{i+1}^n} - \frac{\mathbf{r}_i^{n+1} - \mathbf{r}_{i-1}^{n+1}}{h_i^n} \right) + \frac{\alpha_i^n}{2} (\mathbf{r}_{i+1}^{n+1} - \mathbf{r}_{i-1}^{n+1}), \tag{29}$$

for $i = 1, \dots, m - 1$, with \mathbf{r}_0^{n+1} and \mathbf{r}_m^{n+1} prescribed, by solving which we get the x, y, z coordinates of the grid points representing a new curve position. In order to compute $(\mathbf{N}_v)_i^n$ and α_i^n for Eq. 29 we use the following approach. First we approximate the tangential vector

$$\mathbf{T}_i^n = \frac{\mathbf{r}_{i+1}^n - \mathbf{r}_{i-1}^n}{h_{i+1}^n + h_i^n} \tag{30}$$

and then we compute $(\mathbf{N}_v)_i^n$ by the formula (6) where we assume that the external velocity field \mathbf{v} is given. In our application, it is given as a piecewise constant function in image voxels, so we just have to check in which voxel the current grid point \mathbf{r}_i^n is located. In order to compute α_i^n we integrate the Eq. 24 in the primal segment $[\mathbf{r}_{i-1}^n, \mathbf{r}_i^n]$ and we get

$$\int_{\mathbf{r}_{i-1}^n}^{\mathbf{r}_i^n} \partial_s \alpha \, ds = \int_{\mathbf{r}_{i-1}^n}^{\mathbf{r}_i^n} Uk_1 + Vk_2 - \langle Uk_1 + Vk_2 \rangle_\Gamma + \left(\frac{L}{g} - 1 \right) \omega_r \, ds \tag{31}$$

from where we obtain

$$\alpha_i^n = \alpha_{i-1}^n + h_i^n (U_i^n k_{1i}^n + V_i^n k_{2i}^n) - h_i^n \langle Uk_1 + Vk_2 \rangle_\Gamma^n + \left(\frac{L^n}{m} - h_i^n \right) \omega_r, \tag{32}$$

for $i = 1, \dots, m - 1$, setting $\alpha_0^n = 0$ and getting $\alpha_m^n = 0$. In Eq. 32

$$\langle Uk_1 + Vk_2 \rangle_\Gamma^n = \frac{1}{L^n} \sum_{l=1}^m h_l^n (U_l^n k_{1l}^n + V_l^n k_{2l}^n), \quad L^n = \sum_{l=1}^m h_l^n \tag{33}$$

and the terms $U_i^n, k_{1i}^n, V_i^n, k_{2i}^n$ are approximations of the corresponding quantities in the discrete curve segment $[\mathbf{r}_{i-1}^n, \mathbf{r}_i^n]$ which are obtained as averages of these quantities computed in the grid points \mathbf{r}_{i-1}^n and \mathbf{r}_i^n . In order to get them we first compute the discrete curvature vector

$$(k\mathbf{N})_i^n = \frac{2}{h_{i+1}^n + h_i^n} \left(\frac{\mathbf{r}_{i+1}^n - \mathbf{r}_i^n}{h_{i+1}^n} - \frac{\mathbf{r}_i^n - \mathbf{r}_{i-1}^n}{h_i^n} \right) \tag{34}$$

and then we compute $(\mathbf{N}_1)_i^n, (\mathbf{N}_2)_i^n$ by Eq. 11, k_{1i}^n, k_{2i}^n by Eq. 12 and U_i^n, V_i^n by Eq. 14.

The system of Eq. 29 can be written as

$$\mathcal{A}_i^n \mathbf{r}_{i-1}^{n+1} + \mathcal{B}_i^n \mathbf{r}_i^{n+1} + \mathcal{C}_i^n \mathbf{r}_{i+1}^{n+1} = \mathcal{F}_i^n \tag{35}$$

with coefficients given by

$$\mathcal{A}_i^n = -\frac{\epsilon}{h_i^n} + \frac{\alpha_i^n}{2}, \quad \mathcal{C}_i^n = -\frac{\epsilon}{h_{i+1}^n} - \frac{\alpha_i^n}{2}, \quad \mathcal{B}_i^n = \frac{h_i^n + h_{i+1}^n}{2\tau} - (\mathcal{A}_i^n + \mathcal{C}_i^n),$$

$$\mathcal{F}_i^n = \frac{h_i^n + h_{i+1}^n}{2\tau} \mathbf{r}_i^n + \mu (\mathbf{N}_v)_i^n \frac{h_i^n + h_{i+1}^n}{2}$$

and it represents three tridiagonal systems for the x, y, z coordinates of the grid points representing a new curve position which are solved by the Thomas algorithm which is a fast procedure, numerically stable provided that the system matrix is strictly diagonally dominant. The strict diagonal dominance of the system matrix is equivalent to the condition

$$|\mathcal{B}_i^n| > |\mathcal{A}_i^n| + |\mathcal{C}_i^n| \tag{36}$$

for every $i = 1, \dots, m - 1$ and every n . The term \mathcal{B}_i^n is always positive and if both terms \mathcal{A}_i^n and \mathcal{C}_i^n are less or equal to zero, the condition (36) is clearly fulfilled. The condition (36) could be violated only if $|\frac{\alpha_i^n}{2}| > \frac{\epsilon}{h_i^n}$ but then we can decrease the time step τ in order to keep the strict diagonal dominance of the system matrix and thus guarantee the solvability of the system by the Thomas algorithm. The solvability condition is given by

$$\tau < \frac{1}{2} \frac{h_i^n + h_{i+1}^n}{|\mathcal{A}_i^n| + |\mathcal{C}_i^n| + (\mathcal{A}_i^n + \mathcal{C}_i^n)}, \tag{37}$$

and must be tested for all $i = 1, \dots, m - 1$, where either \mathcal{A}_i^n or \mathcal{C}_i^n is positive. If that happens one has to decrease the time step according to the condition (37). As one can see from Eq. 37, this condition is not restrictive in practice because the time step is

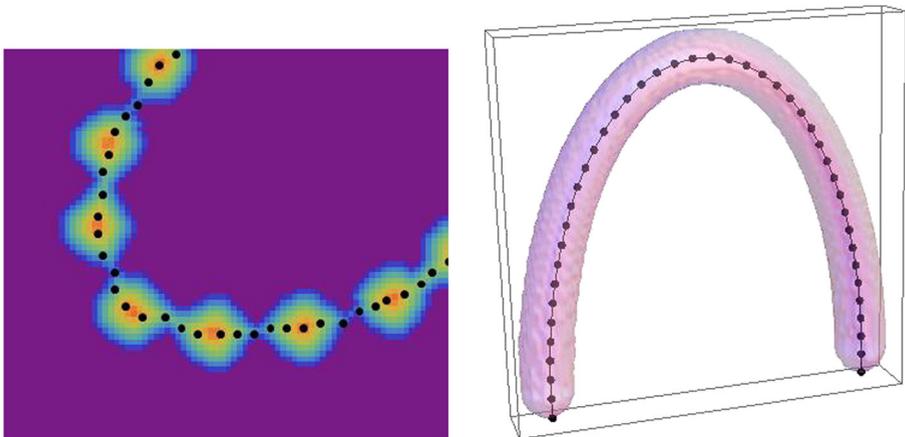


Fig. 11 The results for the test data obtained using the tangential redistribution

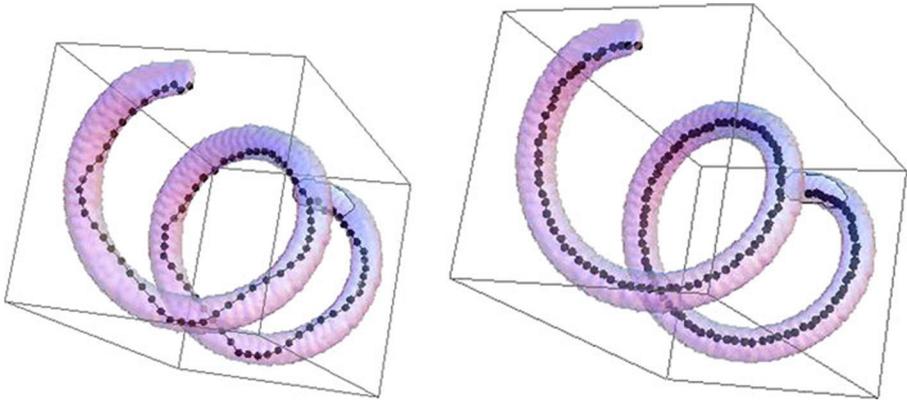
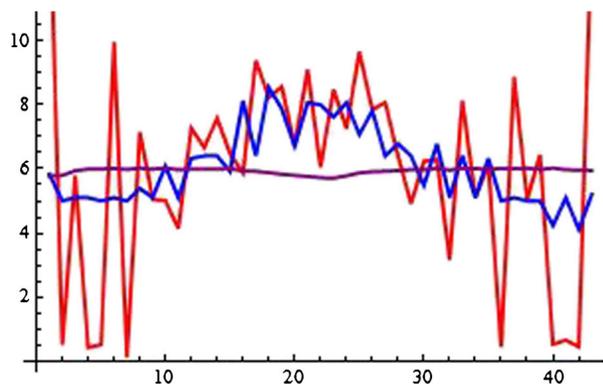


Fig. 12 The results obtained on the helix. The initial (left) and the final (right) path obtained by using the tangential redistribution

just proportional to the spatial discretization step which is uniform along the discrete curve. Although we have to perform additional tests, the numerical solution by the Thomas algorithm is still an efficient procedure and thus the numerical evolution to the steady state is realized in the fast and stable way.

The results for our testing data as well as for the real virtual colonoscopy data are presented in Figs. 11, 12 and 13. In the presented computations we used parameters $\mu = \epsilon = \tau = \omega_r = 1$. The parameter μ gives the velocity of the evolving curve in the normal direction and standardly $\mu = 1$. Then, if we consider the voxel size equal to 1, the standard choice of the time step is $\tau = 1$ (provided that the condition (37) is fulfilled) in order to move the curve not more than one voxel size in one time step. The parameter ϵ influences the smoothness of the curve during the evolution and in the final time, the choice $\epsilon = 1$ is appropriate and any choice close to 1 gives a similar result. The last parameter ω_r controls the speed of tangential redistribution in order to be uniform. Its moderate values, like $\omega_r = 1$, are sufficient to get uniform discretization of the curve during the evolution and in the final state. However, any

Fig. 13 The comparison of the grid point distances: the basic vector field (red), the projected vector field plus the curvature regularization (blue), the final model (24–25) (violet)



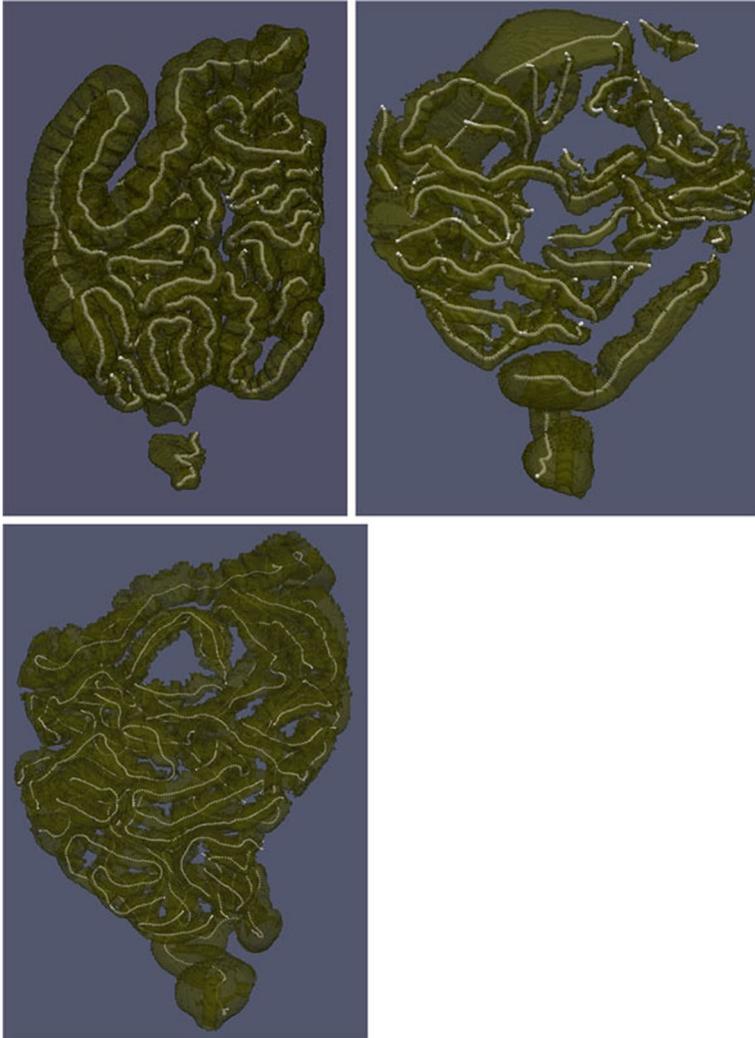


Fig. 14 The results for the real data obtained using the final model (24–25)

choice of $\omega_r \approx O(1)$ is reasonable and yield a comparable result. As one can see, the standard choice of parameters is possible and their slight modification does not alter significantly the final result (Fig. 14).

The illustrative 2D experiments were performed by the method from [17]. Figure 13 shows differences in the grid point distances for our 3D test data (Figs. 8, 10, 11 right), the red curve for the basic velocity field, the blue for the projected vector field plus the curvature regularization and the violet curve for the final model (24–25). In the final model the grid point distances are uniform, the final curve is smoothly centered and can be successively used for the virtual voyage inside the colon.

Acknowledgments This work has been performed in cooperation with the company TatraMed spol s.r.o., Bratislava and supported by the grant APVV-0184-10.

References

1. Barrett, J.W., Garcke, H., Nürnberg, R.: On the variational approximation of combined second and fourth order geometric evolution equations. *SIAM J. Sci. Comput.* **29/3**, 1006–1041 (2007)
2. Barrett, J.W., Garcke, H., Nürnberg, R.: Numerical approximation of gradient flows for closed curves in \mathbb{R}^d . *IMA J. Numer. Anal.* **30**(1), 4–60 (2010)
3. Bovik, A.: *Handbook of Image and Video Processing*. Academic Press, New York (2000)
4. Bourguine, P., Frolkovič, P., Mikula, K., Pyriaras, N., Remešíková, M.: Extraction of the intercellular skeleton from 2D microscope images of early embryogenesis. In: *Lecture Notes in Computer Science* 5567, pp. 39–49. Springer (2009)
5. Dijkstra, E.W.: A note on two problems in connection with graphs. *Numer. Math.* **1**, 269–271 (1959)
6. Dziuk, G.: Convergence of a semi discrete scheme for the curve shortening flow. *Math. Model. Methods Appl. Sci.* **4**, 589–606 (1994)
7. Deckelnick, K.: Weak solutions of the curve shortening flow. *Calc. Var. Partial Differ. Equ.* **5**, 489–510 (1997)
8. Hong, L., Kaufman, A.E., Wei, Y., Viswambharan, A., Wax, M., Liang, Z.: 3D Virtual Colonoscopy. *IEEE Symp. Biomed. Vis.* 26–32 (1995)
9. Hong, L., Muraki, S., Kaufman, A.E., Bartz, D., He, T.: Virtual voyage: Interactive navigation in the human colon. *ACM SIGGRAPH* **97**, 27–34 (1997)
10. Hou, T.Y., Lowengrub, J., Shelley, M.: Removing the stiffness from interfacial flows and surface tension. *J. Comput. Phys.* **114**, 312–338 (1994)
11. Hou, T.Y., Klapper, I., Si, H.: Removing the stiffness of curvature in computing 3-D filaments. *J. Comput. Phys.* **143**, 628–664 (1998)
12. Lefere, P., Gryspeerdt, S. (eds) *Virtual Colonoscopy: A Practical Guide*. Springer-Verlag, Berlin, (2010). ISBN 978-3-540-79879-8
13. Kimura, M.: Numerical analysis for moving boundary problems using the boundary tracking method. *Japan J. Indust. Appl. Math.* **14**, 373–398 (1997)
14. Mikula, K., Ševčovič, D.: Evolution of plane curves driven by a nonlinear function of curvature and anisotropy. *SIAM J. Appl. Math.* **61**, 1473–1501 (2001)
15. Mikula, K., Ševčovič, D.: A direct method for solving an anisotropic mean curvature flow of planar curve with an external force. *Math. Method. Appl. Sci.* **27**, 1545–1565 (2004)
16. Mikula, K., Ševčovič, D.: Computational and qualitative aspects of evolution of curves driven by curvature and external force. *Comput. Visual. Sci.* **6**, 211–225 (2004)
17. Mikula, K., Ševčovič, D., Balažovjeh, M.: A simple, fast and stabilized flowing finite volume method for solving general curve evolution equations. *Comm. Comp. Phys.* **7**(1), 195–211 (2010)
18. Mikula, K., Urban, J.: 3D curve evolution algorithm with tangential redistribution for a fully automatic finding of an ideal camera path in virtual colonoscopy. In: *Proceedings of the Third International Conference on Scale Space Methods and Variational Methods in Computer Vision*, May 29th - June 2nd, 2011, Ein Gedi, Dead Sea, Israel, *Lecture Notes in Computer Science Series*, Springer, 2011
19. Rouy, E., Tourin, A.: Viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.* **29**(3), 867–884 (1992)
20. Sethian, J.A.: *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and material science*. Cambridge University Press, New York (1999)
21. Truyen, R., Deschamps, T., Cohen, L.D.: Clinical evaluation of an automatic path tracker for virtual colonoscopy, medical image computing and computer-assisted intervention (MICCAI). Utrecht, Netherlands (2001)
22. Wan, M., Dachille, F., Kaufman, A.E.: Distance-field based skeletons for virtual navigation. *Proc. IEEE Vis. Conf. VIS'01*, 239–245 (2001)