Extracting Motion Velocities from 3D Image Sequences and Coupled Spatio-Temporal Smoothing

Tobias Preußer and Martin Rumpf [*]

Numerical Analysis and Scientific Computing, Duisburg University, 47048 Duisburg, Germany

ABSTRACT

Recent image machinery delivers sequences of large scale three-dimensional (3D) images with a considerably small sampling width in time. In medical as well as in engineering applications the interest lies in underlying deformation, growth or motion phenomena. A robust method is presented to extract motion velocities from such image sequences. To avoid an ill-posedness of the problem one has to restrict the study to certain motion types, which are related to the concrete application. The derived formulas for the motion velocities clearly reflect the geometry of the motion. Robustness of the presented implementation is based on local regularizations in space-time. Thereby geometric quantities on the image sequences are evaluated on the local regularizations. Examples outline the potential of the proposed method in medical applications (3D ultrasound sequences) and experimental fluid dynamics (3D flow in porous media). As an improved regularization approach an effective denoising method based on anisotropic geometric diffusion for 3D data sets is discussed, which respects important features on levelsets such as edges and corners and accelerated motions and preserves them during the smoothing process. Its application as a pre-processing step turns out to be especially advisable for image sequences with a considerably small signal to noise ratio.

Keywords: image sequences, multiscale, optical flow, anisotropic diffusion

1. INTRODUCTION

Processing three dimensional (3D) images is a task of growing interest in various applications. In medical imaging different image generation hardware such as computed tomography (CT) or magnetic resonance imaging (MRI), and recently also 3D ultrasound (US) devices deliver not only single large scale images but often 3D image sequences. They allow the study of motion, growth, and deformation on small time scales ranging from seconds or minutes up to large time scales of days or years. Furthermore one observes previously hidden temporal physical processes in different media via 3D imaging devices. As an example we mention the flow in porous media where isotope marking is applied to measure a 3D concentration distribution via an MRI device. Again the aim is to study the underlying physical motion and to compare it with model predictions and computed simulations.

The aim of this paper is to present a robust method to extract motion velocity fields from 3D image sequences. Thus, the focus is on the actual extraction and not on the final visualization of the velocity fields. Extracting motion from image sequences is known as the optical flow problem in image processing and has been studied extensively for 2D image sequences.[2, 11, 16, 23, 27] The contribution of this paper is to extend the methodology to 3D, to express the resulting velocities in concise geometric terms, and to focus on an effective and robust implementation. Already a large bunch of literature dealing with visualization of vector fields: Most popular are methods which generate textures aligned to the flow field.[6, 12, 18, 32, 35] The visualization of time-dependent vector fields, which appear in our setting here, has been discussed in.[4, 5, 21, 31] Recently van Wijk[36] presented the IBFV approach, which visualizes vector fields in real time using graphics hardware of modern computers.

In image analysis the interest frequently lies in the extraction of certain level surfaces bounding volumes or specific regions of interest, and their temporal change within the image sequence. Then the actual intensity value is of minor importance and anyway dependent on the modality of the image acquisition process. Therefore it is desired to derive *morphological* methods, which are not affected by a transformation of the intensity. We ask for a motion extraction method which shares this property. At first, we give the classical optical flow equation to extract the normal component of the velocity. In special cases, for instance in case of a physical phenomenon obeying a model like Darcy's law, where the

[*] `[preusser, rumpf]@math.uni-duisburg.de`. The work of Tobias Preusser was funded by the DFG Schwerpunktprogramm 1114.

flux vector is pointing in the direction of the pressure gradient, the normal component completely describes the motion. But in applications such a simplification is not feasible. Unfortunately, the extraction of tangential velocities, turns the task into an ill-posed problem.[27] Thus, the set of possible solution velocities has to be restricted and instead we deal with the apparent velocity[16] — the velocity which arises from a motion locally constant in space. Especially in case of moving objects, or structures with significantly curved details and small variations of the velocity normal to levelsets, the apparent velocity turns out to be meaningful. Alternatively, motivated by modeling in continuum mechanics one can ask for regularizations resulting in deformations controlled by elastic stresses or viscous fluid effects.[7, 9, 15, 19, 20, 34] But this approach is computationally rather expensive and pays off in case of large deformations in between frames of the sequence, which we rule out in our applications here.

Concerning the robust implementation, we apply local L^2 projection onto suitable polynomial spaces in space and time, which can be regarded as a linear filter improving the signal to noise ratio. On these projections we evaluate all geometric quantities necessary to compute the velocity components (cf. Fig. 1) . This approach turns out to be consistent but computationally expensive. We therefore also consider the convolution with derivatives of smoothing kernels as a further linear filtering variant. Typically, the images of a recorded image sequence — especially ultrasound images — are characterized by significant high frequent noise due to measurement errors. Here, linear filtering turns out not to be the appropriate approach. Therefore, we apply a morphological multiscale method for image-sequence processing[26] which combines the image processing methodology based on nonlinear diffusion equations and the theory of geometric evolution problems (cf. Fig. 2). Such nonlinear and anisotropic multiscale methods have proven to be successful tools in image denoising, edge enhancement and shape recovery.[1, 8, 24, 30, 37] Thereby, the image is considered as initial data of a suitable evolution problem. The aim of the method we apply here, is to smooth levelsets of an image while simultaneously preserving geometric features such as edges and corners on the levelsets. This is obtained by an anisotropic curvature evolution.[3, 10, 17, 33] Spatially depended evolution velocities in mean curvature motion models already have been considered by Alvarez et al.[1] and Sapiro,[28] but the model we use here additionally considers the directions of spatial edges and the motion velocity and therefore is anisotropic in space and sequence time.

2. EXTRACTING MOTION (THE SMOOTH CASE)

At first, we will consider a continuous family of images on some time interval $[0, T]$ and derive expressions for the motion velocity in terms of spatial and temporal derivatives of this continuous image sequence. Let us denote the continuous image sequence with $\phi : [0, T] \times \Omega \to \mathbb{R}, (s, x) \mapsto \phi(s, x)$. Here $\Omega := [0, 1]^d$ is the image domain in the d-dimensional Euclidean space for $d = 2$ or $d = 3$ (which we assume to be fixed) and s is the sequence time parameter. We will always denote sequence parameters by r and s respectively, whereas x and y respectively stand for the spatial coordinates. We assume ϕ to be smooth in time and space. The perspective of levelsets on images of the sequence will play a central role in our motion analysis. We denote by $\mathcal{M}^c(s)$ the levelset — the isosurface — of $\phi(s, \cdot)$ corresponding to the level value $c \in \mathbb{R}$, i. e. $\mathcal{M}^c(s) = \{x \in \Omega \,|\, \phi(s, x) = c\}$. For the sake of simplicity we will skip the index c indicating the level value and the sequence parameter s whenever they are clear from the context. If the image is not locally flat, the normal $N(s, x)$ at x in the image $\phi(s, \cdot)$ is given by $N(s, x) = |\nabla\phi(s, x)|^{-1}\nabla\phi(s, x)$. The tangent space at x on $\mathcal{M}$ is denoted by $\mathcal{T}_x\mathcal{M} = \mathrm{span}\{N\}^\perp$.

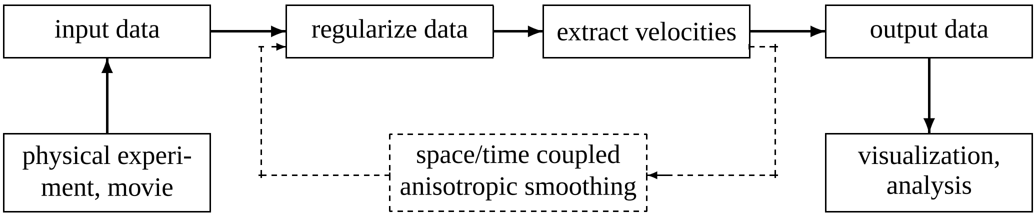

Figure 1. As input data the process takes a 2D respectively 3D image sequence, resulting e.g. from the observation of a physical experiment. Due to the insufficient regularity of the data it has to be regularized before the extraction of the motion quantities. The final output is a vector field, describing the motion of the temporally changing level-sets of the sequences. As a sophisticate alternative to the linear regularization, one can apply a spatio/temporal coupled anisotropic smoothing process, which denoises the data, but keeps significant spatial edges and temporal accelerations (dashed lines).

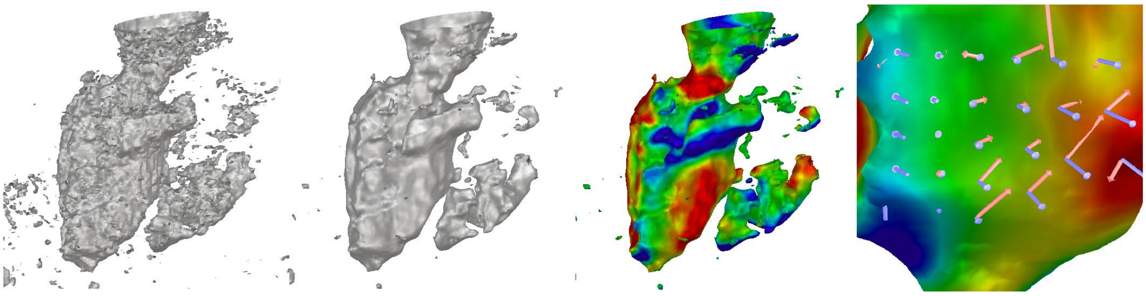

Figure 2. The noisy initial frame (left) is pre-processed by an an anisotropic geometric smoothing method. On the smoothed data (middle left) we compute the motion of the levelsets (middle right). A color ramp from blue (moving inward) to red (moving outward) indicates the normal component of the velocity. On a magnified section (right) we have depicted the splitting of v in tangential (red arrows) and normal (blue arrows) components. All computations were performed on a 128^3 grid.

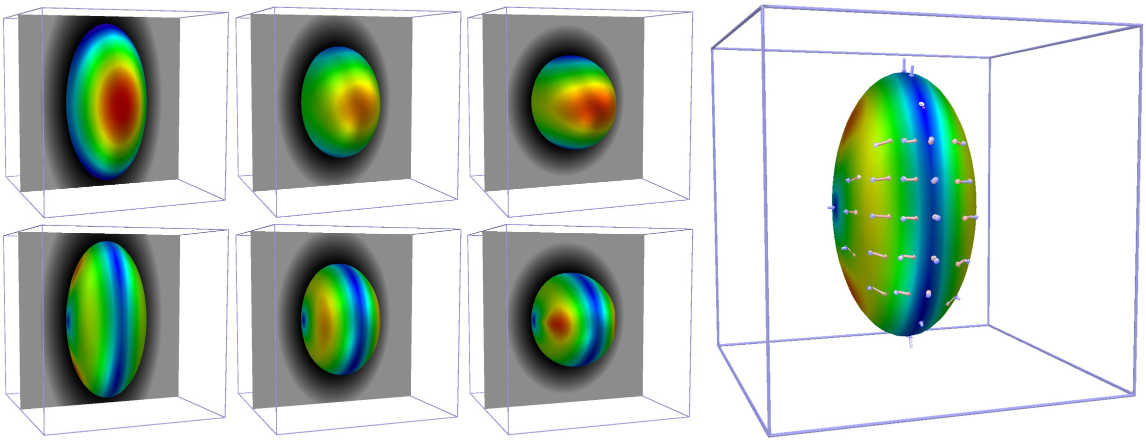

Figure 3. As a test case we extract the motion from the evolution of ellipsoidal levelsets with oscillating half axes. I.e. we consider the image sequence $\phi(s, x_1, x_2, x_3) := x_1^2/a(s) + x_2^2/b(s) + x_3^2$, where $a(s) := 4s - (1 - s)$, $b(s) := s - 4(1 - s)$ for $s \in [0, 1]$. We have depicted the results of the velocity computation on the same levelset (isosurface) in different frames of the sequence. In the upper row a color ramp from blue (moving inward) to red (moving outward) indicates the normal component of the velocity. In the lower row the color ramp from blue to red indicates the absolute value of the tangential component of the velocity. The large picture on the right shows the decomposition of v in its tangential and normal part.

Now let us suppose that there is a correspondence between the separate time slices of the image sequence due to a physical motion. This motion influences the image intensity we observe in sequence time. Suppose $v : [0, T] \times \Omega \to \mathbb{R}^d, (s, x) \mapsto v(s, x)$ is the velocity field generating the motion in space and time. Thus, for a single motion trajectory $\{x(s)\}$ running through a point $x_0 \in \mathcal{M}^c(s_0)$ on some levelset $\mathcal{M}^c(s_0)$, we obtain

$$x(s_0) = x_0\,, \qquad \dot{x}(s) = v(s, x(s)).$$

Clearly any velocity vector $v(s_0, x_0) \in \mathbb{R}^3$ can be decomposed into a normal part $v_\mathrm{n} \in \mathrm{span}\{N\}$ and a tangential part $v_\mathrm{tg} \in \mathcal{T}_x\mathcal{M}$, such that $v = v_\mathrm{n} + v_\mathrm{tg}$ and $v_\mathrm{n} = (v \cdot N)N$.

2.1. Normal velocity

Our first assumption on the image sequence is now, that intensities are preserved along motion trajectories. This is reasonable because intensities in image sequences resulting from physical experiments or medical devices are usually functions of certain physical quantities like density or concentration. If these quantities move in space, so do the image intensities. In case of moving solid objects with certain intensities the assumption is even more obvious. Thus, we get

$$\phi(s_0 + \tau, x(s_0 + \tau)) = \phi(s_0, x_0) \qquad \forall \tau \in [-s, T - s]\,. \tag{1}$$

We differentiate this with respect to τ and obtain $\partial_s\phi(s,x) + \nabla\phi(s,x)\cdot v(s,x) = 0$, where "$\cdot$" always denotes the scalar product. Assuming $|\nabla u| \neq 0$ and dividing by $|\nabla\phi|$ we end up with an expression for the normal component of the velocity:

$$v_{\mathrm{n}} = (v\cdot N)N = -\frac{\partial_s\phi}{|\nabla\phi|}N.$$

Based on assumptions on the underlying physics of the phenomenon recorded in the image sequence we may know that the motion velocity is expected to be normal to intensity levels (see Section 4). In this case we have $v = v_{\mathrm{n}}$ and the above equation completely describes the motion. But in general we can not expect the tangential component v_{tg} of the velocity to vanish.

2.2. Apparent velocity

Let us assume that the image sequence consists of fixed objects moving in space but not being deformed. Hence, we consider a motion velocity $v(s,x)$ which is constant in space, i. e. $v(s,x) = v(s)$. Then obviously all levelsets are just translated. Therefore, not only the intensities $\phi(s,x)$ but also the normals $N(s,x)$ on the levelsets stay constant along motion trajectories:

$$N(s_0+\tau, x(s_0+\tau)) = N(s_0,x_0) \qquad \forall\tau\in[-s,T-s]\,.$$

We differentiate this with respect to τ at $\tau = 0$, and get an expression which involves the Jacobian ∇N of the normal:

$$\frac{\partial}{\partial\tau}N(s+\tau,x(\tau))\Big|_{\tau=0} = 0 \qquad\qquad \Rightarrow \qquad\qquad \partial_s N(s,x) + \nabla N(s,x)\,v(s,x) = 0. \tag{2}$$

Before we continue, let us more closely examine the Jacobian $S := \nabla N$ of the normal, which is the projection of the second derivative $D^2\phi$ onto the tangent space $\mathcal{T}_x\mathcal{M}$. Denoting the tensor product $u\otimes v = (u_i v_j)_{ij}$, we have[25, 26]

$$S = \nabla N = \frac{1}{|\nabla\phi|}(\mathrm{Id} - N\otimes N)D^2\phi. \tag{3}$$

The mapping S can be regarded as an extension of the shape operator $S_{\mathcal{T}_x\mathcal{M}}$ on the levelset[8, 13] and has eigenvalues $\{\kappa^1,\kappa^2,0\}$ corresponding to eigenvectors $\{v^1,v^2,N\}$. Indeed we have $S_{\mathcal{T}_x\mathcal{M}} = S(\mathrm{Id}-N\otimes N)$. Furthermore κ^1,κ^2 turn out to be the principal curvatures and v^1 and v^2 are the corresponding principal directions of curvature. From $\nabla N N = 0$ we deduce that the orthogonal complement $\mathcal{T}_x\mathcal{M}^\perp = \mathrm{span}\{N\}$ of the tangent space lies in the kernel of S and from $0 = \partial_s(N\cdot N) = 2\partial_s N\cdot N$ we observe that $\partial_s N \in \mathcal{T}_x\mathcal{M}$. Thus, we can rewrite Eq. (2) as an equation on the tangent bundle: $\partial_s N(s,x) + S_{\mathcal{T}_x\mathcal{M}}v_{\mathrm{tg}} = 0$. If the shape operator $S_{\mathcal{T}_x\mathcal{M}}$ is invertible one obtains an expression for the tangential component of the velocity

$$v_{\mathrm{tg}} = -S^{-1}_{\mathcal{T}_x\mathcal{M}}\partial_s N\,.$$

Finally adding the normal and the tangential part, we have derived the *apparent velocity*

$$v_{\mathrm{app}} = v_{\mathrm{tg}} + v_{\mathrm{n}} = -S^{-1}_{\mathcal{T}_x\mathcal{M}}\partial_s N - \frac{\partial_s\phi}{|\nabla\phi|}\,N\,. \tag{4}$$

For two dimensional image sequences, this expression was derived by Guichard,[16] although he did not express it with these geometric terms. In our derivation, all geometrical quantities intrinsically describe levelsets in spaces of arbitrary dimension. Thus (4) is applicable in n-dimensional space and especially for $n = 3$. Let us emphasize that this velocity depends on the morphology of images only. Indeed, it is obviously invariant under monotone gray value transformations $\tilde\phi(s,x) := \beta\circ\phi(s,x)$.

2.3. General motion

In general an observed motion does not have the simple structure of a spatially constant motion as assumed in the last section. Nevertheless, an application of the above definition (4) for motions which are not constant in space still can make sense. In fact, if an image is locally rather complex — the levelsets are significantly curved — and the motion on this

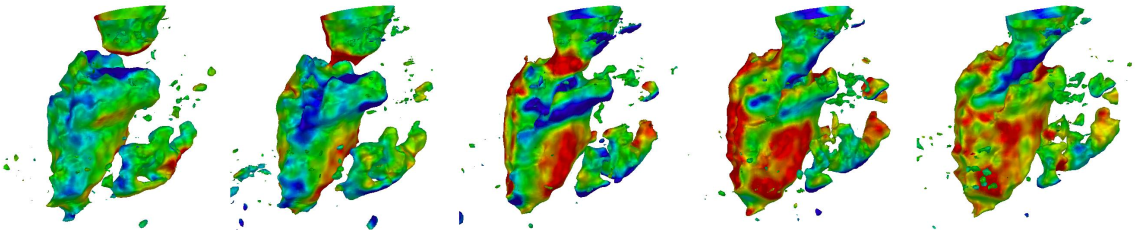

Figure 4. From left to right we show the extracted velocity on several frames from the sequence of echocardiographical images of the human heart (cf. Fig. 2). Again the color ramp from blue (moving inward) to red (moving outward) codes the normal component of the velocity. The computations were performed on a 128^3 grid.

image region is close to being constant, then we expect the above formula to give a good approximation of the actual velocity.

Concerning the tangential part of the velocity one easily sees that its extraction is an ill posed task: Suppose v describes the motion of the levelsets and let w be an arbitrary vector field such that $w(s,x) \in \mathcal{T}_x\mathcal{M}$. Obviously the motion described by w results in tangential movements of the levelsets within themselves, and thus can not be recognized by the observer. Therefore $v + w$ also describes the motion of the levelsets, but w can be arbitrarily irregular. The apparent velocity selects from the set of all possible solutions the one, which minimizes the variation of the normal along the trajectory, i.e. $\partial_\tau N(s_0 + \tau, x(s_0 + \tau)) = 0$ (cf. Fig. 5).

Concerning the restriction to configurations with an invertible shape operator, one may ask whether this is a shortcoming of our formula or if there is one more limitation of the motion capturing itself. Hence, let us consider an image whose levelsets are concentric (infinitely extended) cylinders and this set of cylinders is supposed to move in space generating an image sequence. On one hand the velocity component along the axis of the cylinder obviously can not been extracted from the image sequences. On the other hand this direction is a principal direction of curvature with curvature value 0 (cf. Fig. 5). If a levelset is flat in one direction at least up to second order ($\kappa^i = 0$ for some $i \in \{1,2\}$), then we are locally in the cylindrical case and reach the limits of the motion capturing.

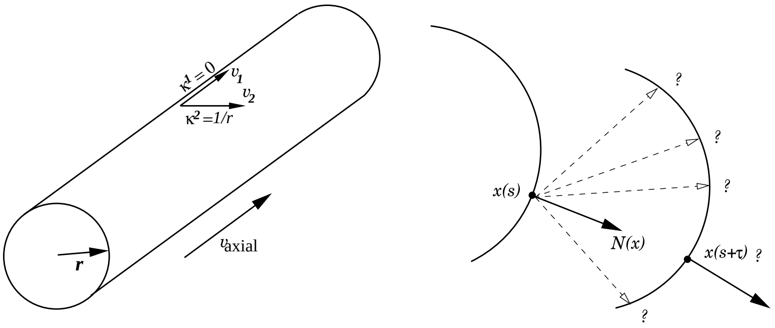

Figure 5. Left: In case of a moving cylinder the velocity component along the axis of the (infinitely extended) cylinder can not be identified. The image sequence is actually invariant under motions in the axial direction. This corresponds to the observation that the axial direction is a principle curvature direction with curvature 0. Right: From the set of all possible solutions, the apparent velocity minimizes the variation of the normal.[16]

3. A ROBUST IMPLEMENTATION

In the formulas to compute the apparent velocity v_{app} we have made use of geometric quantities which involve higher order derivatives in space and time. Typically our image sequences are given as a sequence of temporally equidistant frames – we denote the temporal distance between the frames with Δs. The image frames usually are given in voxel or multilinear finite element form. Furthermore, they are overlaid with some noise resulting from measurement errors. Thus, we have to robustly regularize the measurement results coded in the image sequence, to make the extraction process more insensitive to noise. Moreover we have to explain how to robustly evaluate derivatives on this low regularity data. In principle, there are different approaches at hand.[25] We will consider two different variants here, which combine the aspects of regularizing the data and defining higher order derivatives on constant or at most piecewise linear data. First, we focus on a local projection of the image onto a polynomial space, and secondly we will describe an approach which uses

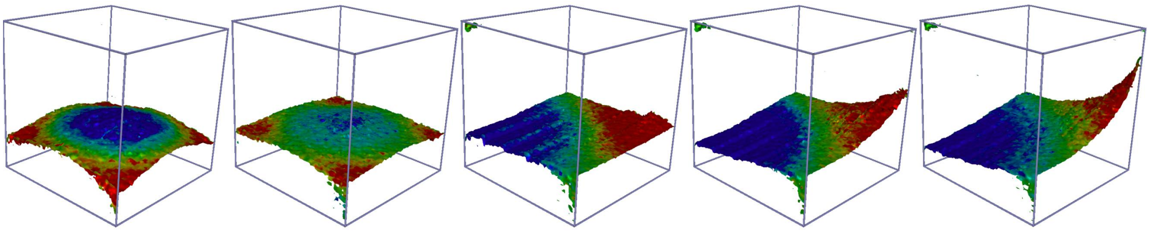

Figure 6. The application of the method to experimental data from a fresh and salt water experiment is depicted. First, salt water is floating into a container filled with fresh water. Then an outlet is opened at the top left corner of the container. During the experiment, the salt-concentration was measured using an MR imaging device. From left to right several frames of the experiment are shown. A color ramp again indicates the normal component of the velocity (cf. Fig. 4). The computation was performed on a 64^3 grid.

convolutions with smoothing kernels and their derivatives.[26] During the expositions we will indicate quantities resulting from a regularization process by a superscript σ.

In the derivation presented in the previous sections, we have always assumed the definition of a normal was possible, and moreover we assumed the shape operator $S_{\mathcal{T}_x\mathcal{M}}$ being invertible. Obviously, for real data the gradient $\nabla\phi$ could vanish, and we can not divide by $|\nabla\phi|$ any more. Therefore we replace[14] the Euclidean norm by $|\nabla\phi|_\epsilon := \sqrt{|\nabla\phi| + \epsilon^2}$ for a small ϵ. Moreover, if a levelset is flat at least in one direction (i.e. $\kappa^i = 0$ for some $i \in \{1,2\}$) we replace the inverse $S^{-1}_{\mathcal{T}_x\mathcal{M}}$ of the shape operator by its pseudo-inverse $S^{\dagger}_{\mathcal{T}_x\mathcal{M}}$, which only inverts $S_{\mathcal{T}_x\mathcal{M}}$ on the subspace where it is invertible. On the subspace on which $S_{\mathcal{T}_x\mathcal{M}}$ is not invertible the pseudo-inverse $S^{\dagger}_{\mathcal{T}_x\mathcal{M}}$ is trivially set to zero.

3.1. Regularization via Local Projections

We base the local regularization on a local L^2 projection of the data ϕ onto space $\mathcal{Q}^*$ of sufficient smoothness. Since we need first order derivatives in sequence time and second order derivatives in space, we choose $\mathcal{Q}^*$ to contain tensor products of first order polynomials in sequence-time and second order polynomials in space. Then we will replace the derivatives of the image in the computation of v_{app} by the derivatives of the projection. To this end for the projection we take all monomials into account that give a contribution to the desired derivatives. Thus, we consider

$$\mathcal{Q} := \operatorname{span}\{y_1^2, y_2^2, y_3^2,\ y_1y_2, y_1y_3, y_2y_3,\ y_1, y_2, y_3,\ ry_1, ry_2, ry_3,\ r\} \subset Q^*. \tag{5}$$

Without loss of generality let us now fix a sequence parameter $s = 0$, a point $x = 0$ and a small neighborhood $\mathcal{B}_\sigma(s,x)$ of (s,x). Denoting the basis functions in (5) with $\{q_i\}_i$, and considering a representation $\phi^\sigma_{(s,x)}(r,y) := \phi^\sigma_{(s,x)}(r,y_1,y_2,y_3) := \sum_{i=1}^{13} \alpha_i q_i(r,y_1,y_2,y_3)$ for $\alpha := (\alpha_i)_i \in \mathbb{R}^{13}$ the local projection $\phi^\sigma_{(s,x)}(r,y)$ is defined such that it fulfills the orthogonality relation (=linear system of equations)

$$H\alpha = R \qquad \text{where } R = \left(\int_{\mathcal{B}_\sigma(s,x)} \phi(r,y)\, q_i(r,y)\; dr\, dy\right)_i \quad \text{and} \quad H = \left(\int_{\mathcal{B}_\sigma(s,x)} q_i(r,y)\, q_j(r,y)\; dr\, dy\right)_{ij}. \tag{6}$$

Based on the coefficients α_i of the projection $\phi^\sigma_{(s,x)}(r,y)$ we can compute the desired derivatives and evaluate them in $(r,y) = (0,0)$ to obtain

$$\begin{aligned} \nabla_y \phi^\sigma_{(s,x)}\Big|_{(r,y)=(0,0)} &= (\alpha_7, \alpha_8, \alpha_9)^T, & D^2_y \phi^\sigma_{(s,x)}\Big|_{(r,y)=(0,0)} &= \begin{pmatrix} 2\alpha_1 & \alpha_4 & \alpha_5 \\ \alpha_4 & 2\alpha_2 & \alpha_6 \\ \alpha_5 & \alpha_6 & 2\alpha_3 \end{pmatrix}, \\ N^\sigma_{(s,x)}\Big|_{(r,y)=(0,0)} &= \frac{1}{\sqrt{\alpha_7^2+\alpha_8^2+\alpha_9^2}} \begin{pmatrix} \alpha_7 \\ \alpha_8 \\ \alpha_9 \end{pmatrix}, & \partial_r N^\sigma_{(s,x)}\Big|_{(r,y)=(0,0)} &= \frac{(\mathbb{1} - N^\sigma \otimes N^\sigma)}{\sqrt{\alpha_7^2+\alpha_8^2+\alpha_9^2}} \begin{pmatrix} \alpha_{10} \\ \alpha_{11} \\ \alpha_{12} \end{pmatrix}, \end{aligned} \tag{7}$$

and finally $\partial_r \phi^\sigma_{(s,x)}|_{(r,y)=(0,0)} = \alpha_{13}$. Moreover we obtain S^σ from (3), and then $S^\sigma_{\mathcal{T}_x\mathcal{M}^\sigma}$ by restricting S^σ to the tangent space $(N^\sigma)^\perp$. Substituting these quantities into (4) we get the regularized apparent velocity v^σ_{app}. This regularization approach delivers a consistent way of computing curvature on piecewise linear data.[25] But unfortunately it is computationally very expensive, since it involves a lot of integrations on the neighborhoods $\mathcal{B}_\sigma$.

3.2. Regularization via derivatives of convolution kernels

The regularization presented in the following paragraph is much cheaper in terms of computational costs. For a fixed $\sigma \in \mathbb{R}$, we take the C_0^∞ kernel $K(s,x) := Z^{-1}\exp((|x|^2+s^2)/(|x|^2-s^2-\sigma^2))$ having compact support and use the convolution property for any spatial or temporal derivative D^γ

$$D^\gamma\left(K*\phi\right)(s,x) = D^\gamma\left(\int K(r,y)\phi(s-r,x-y)\,dr\,dy\right) = K*D^\gamma\phi(s,x) = D^\gamma K*\phi(s,x). \tag{8}$$

The constant Z is chosen such that $\int K = 1$ and again σ is a measure for the support of K and — for larger σ the regularization will be stronger. For the computation of the velocity v_{app} we replace all derivatives by convolutions with the corresponding derivatives of the smoothing kernel $D^\gamma K$. According to the convolution property (8) we thus obtain the derivatives of data ϕ smoothed with K:

$$\begin{aligned} N^\sigma &:= \frac{\nabla K * u}{|\nabla K * u|_\epsilon}(s,x), & (\partial_s N)^\sigma &= (\partial_s K * N^\sigma)(s,x) \\ S^\sigma &= \frac{1}{|\nabla K * u|_\epsilon}(\mathrm{Id} - N^\sigma \otimes N^\sigma)(D^2K*u)(s,x), & (\partial_s\phi)^\sigma &= (\partial_s K * \phi)(s,x) \end{aligned} \tag{9}$$

Again we obtain the shape operator $S^\sigma_{\mathcal{T}_x\mathcal{M}}$ by restricting S^σ to the tangent space $\mathcal{T}_x\mathcal{M}$ and the regularized apparent velocity v^σ_{app} from (4).

3.3. Implementation

Let us now focus on the implementational aspects of the regularization approaches we have discussed before. Since we assume the frames of the sequence being given on quadtree respectively octtree grids, it is natural to define σ in multiples of the gridwidth h.

Projection approach. To keep the computational effort moderate, we confine ourselves with two additional sequence frames in the definition of $B_\sigma(s,x)$: For $l \in \{2m \mid m \in \mathbb{N}\}$, we define $B_\sigma(s,x)$ as the patch containing $(l+1)\times(l+1)\times(l+1)$ nodes of each of the frames $\Phi(s-\Delta s,\cdot)$, $\Phi(s,\cdot)$, and $\Phi(s+\Delta s,\cdot)$ such that (s,x) is in the center of the patch. We relate σ to the spatial extend of $\mathcal{B}_\sigma$ by setting $\sigma := lh$.

On $\mathcal{B}_\sigma$ we now perform the integration of the right hand side of (6), by splitting the integral into a sum of integrals on the quadtree/octtree-elements $E \subset \mathcal{B}_\sigma$. On each element E we thereby use a tensor product quadrature rule, with two nodes in each spatial coordinate direction and a simple trapezoidal rule in the sequence-time direction. Having the image shifted locally such that $(s,x) = (0,0)$ and $\phi(s,x) = 0$ we can precompute the inverse of the matrix H in advance, since it does not change with (s,x) and ϕ for a fixed stencil with σ. This speeds up the computation significantly in an implementation. We proceed with substituting the coefficients α_i into the formulas (7) and (4) from which we proceed with simple computations to obtain the regularized apparent velocity v^σ_{app} via the definition (4).

Let us remark here that for a single frame the formula for computation of the shape operator — which is needed to evaluate curvature of levelsets — does not change.[25] Although the matrix to compute the projection is then 9×9. For a comment on the optimal choice of σ we refer the reader to an earlier work[25] of the authors.

Convolution approach. For a fixed σ, we assume the image again to be translated such that $(s,x) = (0,0)$, Then the kernel K has support in $\mathcal{B}_\sigma(s,x) := [-\sigma,\sigma]^{d+1}$. As usual we replace the convolutions with the derivatives D^γ of K with a weighted summation

$$(D^\gamma K*\phi)(s,x) = \int_{\mathcal{B}_\sigma} D^\gamma K(r,y)\phi(s-r,x-y)\,dr\,dy \approx \sum_{E\subset\mathcal{B}_\sigma}\phi(s_E)\int_E D^\gamma K(r,y)\,dr\,dy$$

over the values of ϕ at the center s_E of the involved elements E. The weights are thus obtained by integrating the derivatives $D^\gamma K$ over the elements $E\subset\mathcal{B}_\sigma(0,0)$ and therefore can be precomputed in advance for a fixed σ, which again speeds up the computation significantly. We proceed as before with substituting the convoluted quantities into (9) and (4) from which we again obtain the regularized apparent velocity v^σ_{app}.

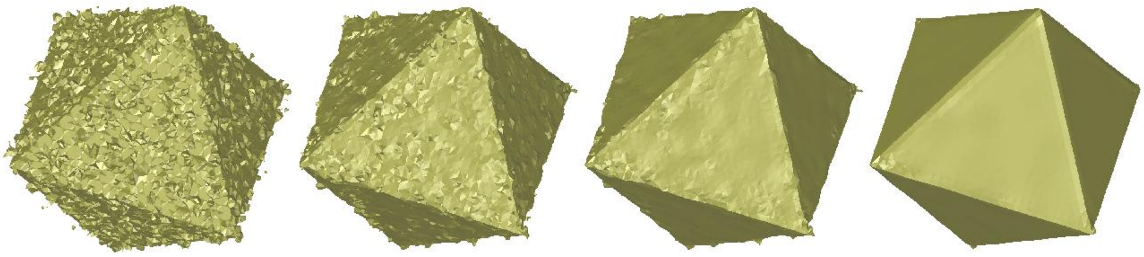

Figure 7. As a test case we consider the function $\phi(x) = |x_1| + |x_2| + |x_3|$ whose levelsets are octahedrons. This function was perturbed and then taken as initial data for the anisotropic geometric diffusion method. From left to right an original perturbed levelset and the corresponding first, second, and fifth time step of its evolution on a 64^3 grid are visualized extracting levelsets.

4. APPLICATIONS

Understanding of the physical phenomenon of a process delivering an image sequence may give more information on the splitting of the velocity in normal and tangential component. In Fig. 4 we have considered a sequence acquired by echocardiography of the human heart. The whole sequence consists of one complete cardiac cycle, in which the heart fills itself with blood and pumps it out again. From the physical point of view this is an elastic deformation.

In many cases one can obtain further information directly from the underlying physics. If we e.g. consider the flow in a porous medium (cf. Fig. 6), we know that Darcy's law applies: The flow of matter will always be in direction of the pressure gradient (here, = normal N to the levelset). Thus, the velocity will have no tangential component and it suffices to consider the normal component. Other physical contexts (e.g. Fick's law, presence of a potential flow) may also apply.

Unavoidably the observations of physical processes via electronic image acquisition devices carry measurement errors resulting in low signal quality. Especially in medical imaging ultrasound- and low dose computed tomography-imaging suffer from high frequent noise (cf. Fig. 2, left). The robustness of the method presented so far relies on the fact that the L^2 projection onto quadratic polynomials on a stencil B_σ respectively the convolution can be regarded as a linear filter. With increasing parameter σ the damping of noise will be intensified. As in most image processing applications linear filters are pretty good in the damping of noise but simultaneously they are known to destroy important features of the levelsets, as corners and edges. Instead one can apply nonlinear multiscale filtering techniques, as they already proved successful in various image processing applications.[1, 22, 24, 29, 37] In the next section, we will consider more sophisticate regularization variants,[25, 26] which have a nonlinear and anisotropic behavior. The results (cf. Fig. 2, middle left) then can be used to extract better regularized apparent velocities (cf. Fig. 2, middle right and Fig. 1).

5. FAIRING BY ANISOTROPIC GEOMETRIC DIFFUSION

5.1. A steady image approach

We consider here a morphological levelset method for the pre-processing of noisy 3D images from sequences before we evaluate velocity terms (with a rather small stencil width σ). It is closely related to the anisotropic geometric diffusion method[8] for the fairing of triangulated surfaces. Indeed, we consider eigenvalues and eigendirections of the shape operator $S_{\mathcal{T}_x\mathcal{M}}$ as feature indicators in a diffusion process. In the levelset formulation we simultaneously deal with all levelsets — hence we actually process a complete 3D image. Up to now this nonlinear smoother works only on a single frame from the 3D image sequence and does not correlate different time steps appropriately. An extension being presented in section 5.2 takes into account the whole image sequence. Let us fix a sequence time s_0 and consider the correponding frame $\psi(x) := \phi(s_0, x)$. Since we assume ψ and thus the levelsets of ψ to be noisy, we seek for a family of successively filtered intensity functions $\{\psi_t(\cdot)\,|\,t \in \mathbb{R}^+\}$ with $\psi_t(\cdot) : \Omega \to \mathbb{R}$. The parameter t serves as the scale parameter as common in multiscale calculus on images and surfaces and $\psi_\bullet(\cdot)$ obeys the following partial differential equation

$$\partial_t \psi - |\nabla\psi| \operatorname{div}\left(a^\sigma\, |\nabla\psi|^{-1} \nabla\psi\right) = 0\,.$$

given $\psi_0(\cdot) = \phi(s_0, \cdot)$. For $a^\sigma = \mathbb{1}$ we obtain an evolution of all levelsets by mean curvature motion, which is the most natural pure smoothing process on surfaces.[17] Our method operates more sophisticate to preserve not only important levelsets (2D edge-surfaces) but also essential features on these levelsets, such as 1D edge-lines. Indeed we assume a^σ to depend on a regularized shape operator S^σ (cf. Section 2.2), where we consider the same regularizations as above. In abstract notation we choose $a^\sigma = (B^\sigma)^T G(S^\sigma_{\mathcal{T}_x\mathcal{M}}) B^\sigma$ where B^σ is the basis-transformation from the frame $\{v^{1,\sigma}, v^{2,\sigma}, N^\sigma\}$ of principal directions of curvature to the standard Euclidean basis and $G(s) = (1 + s^2\lambda^2)^{-1}$ for some positive constant λ here applied to the endomorphism $S_{\mathcal{T}_x\mathcal{M}}$. In the basis $\{v^{1,\sigma}, v^{2,\sigma}, N^\sigma\}$ we therefore obtain $a^\sigma = \operatorname{diag}(G(\kappa^{1,\sigma}), G(\kappa^{2,\sigma}), 0)$, where $\kappa^{1,\sigma}, \kappa^{2,\sigma}$ are the corresponding regularized principal curvatures. Thus, in the direction of a large principal curvature — which is expected to indicate the direction perpendicular to an edge — we apply a significant damping of the diffusion to preserve this edge feature. On the other hand in a orthogonal direction frequently corresponding to a smaller curvature value we allow for a tangential smoothing along the edge. Further details on the method and a finite element implementation can be found in a previous paper by the authors.[25] Figs. 2 and 7 show results obtained by this smoothing method.

5.2. Fairing of image sequences

Still the anisotropic levelset method presented above does not take into account the sequence direction. Thus, there is in general no correlation between successive frames although the original data is temporally correlated, being the observation of a temporal process. We consider an extension of the latter model, which results in a coupled spatio-temporal problem.[26] Additionally to the eigenvalues and eigendirections of the shape operator on levelsets $S_{\mathcal{T}_x\mathcal{M}}$ as feature indicators in a diffusion process, we moreover consider the apparent direction of motion v_{app} of the levelsets and their apparent acceleration $\text{accel}^\sigma := \partial_s v^\sigma_{\text{app}}$ as the driving forces for the diffusion process. We end up with a multiscale of image sequences $\psi_t : [0,T] \times \Omega \to \mathbb{R}$ for $t \in \mathbb{R}^+$, which obeys the following (d+1)-dimensional PDE problem:

$$\partial_t \psi - |\nabla_{(s,x)}\psi| \operatorname{div}_{(s,x)} \left(a^\sigma_{(s,x)} |\nabla_{(s,x)}\psi|^{-1} \nabla_{(s,x)}\psi \right) = 0$$

with the initial condition $\psi_0(s,x) = \phi(s,x)$. Here $\nabla_{(s,x)}$ denotes the spatio-temporal gradient (∂_s, ∇) and $\operatorname{div}_{(s,x)} := \partial_s + \nabla$ its dual operator. In the spatial direction, the diffusion tensor $a^\sigma_{(s,x)}$ still enables for a good preservation of edges as in the above model, but in direction of v_{app} the diffusion will depend on the apparent acceleration accel^σ of the levelsets:

$$a^\sigma_{(s,x)} = a^\sigma_v \begin{pmatrix} 1 \\ v^\sigma_{\text{app}} \end{pmatrix} \otimes \begin{pmatrix} 1 \\ v^\sigma_{\text{app}} \end{pmatrix} + \left(\begin{array}{c|c} 0 & \\ \hline & a^\sigma_{\mathcal{T}_x\mathcal{M}} \end{array} \right),$$

where $a^\sigma_v = G(|\partial_s v_{\text{app}}|)$. This choice of the diffusion tensor will enable for a good preservation of the temporal coupling of the levelsets via the underlying motion phenomenon. The linear system resulting from a finite element discretization and an operator splitting scheme will be solved by a symmetric block solver.[26]

6. CONCLUSIONS

We have presented a robust method to extract motion information from 3D image sequences. Formulas to compute the normal and the apparent velocity have been presented in concise geometric terms and an analysis of the relation of a general motion field and the apparent velocity is given in the appendix. Different applications demonstrate the effectiveness and robustness of the method. The focus of this paper was not to derive a new visualization of a given motion field but to demonstrate the possibility to extract such fields from rather noisy image sequences as they are frequently observed in experimental settings. In addition we considered a nonlinear multiscale filtering for steady images as well as image sequences as an often indispensable pre-processing step. The core of the methods are anisotropic geometric diffusion processes in levelset formulation, driven by geometrical features like edges and corners of the levelset and their acceleration in sequence-time. Interesting future research directions (partially already in progress) are to study the physical role of the apparent velocity in different applications, and to ask for assumptions on physical models underlying the observed motion, which in more cases may help to compute a good approximation of the true physical velocity from the image sequence.

More information about the project including color figures and more examples can be found on our web page

`http://www.numerik.math.uni-duisburg.de/exports/velocity/index.html`

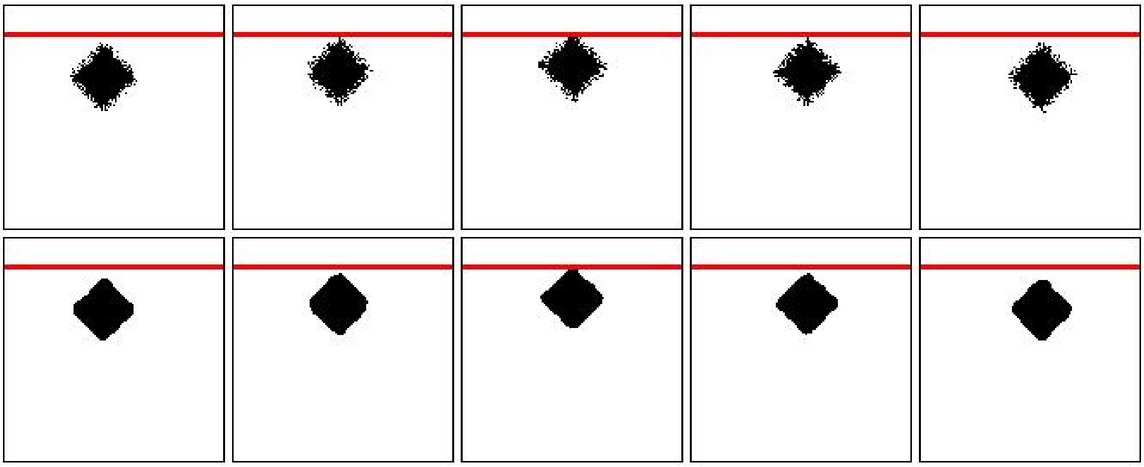

Figure 8. For a continuous test image sequence showing a square bouncing at an invisible object (indicated by the red line), i.e. $\phi(s,x) = |x + d(s))|_1$ where $d(s)$ is the motion of the center of the square, we show the evolution under the coupled spatio temporal diffusion. From left to right several frames near the accelerated point of the motion (= the square bounces) are depicted. The upper row shows the noisy image sequence, whereas in the lower row the sequence resulting from the third scale step of the evolution is shown. Clearly the shape is smoothed significantly but its corners are preserved.

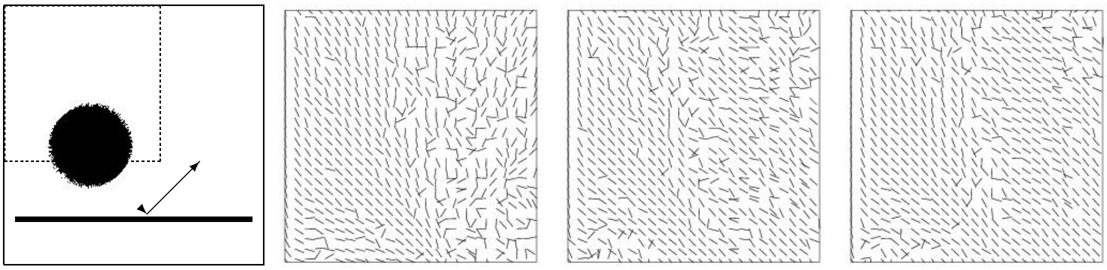

Figure 9. We consider a noisy continuous test sequence showing a ball bouncing at some invisible object (left) indicated by the thick line, i.e. $\phi(s,x) = |x + d(s)|_2$, where $d(s)$ is the motion of the center of the ball. The improvement of the extracted velocities during the evolution are depicted. From middle left to right the extracted velocities of the noisy data and scale steps 1 and 3 are depicted. We have drawn the normalized velocities for always the same frame of the sequence and the region indicated by the dashed area in the image left. Clearly the extracted motion becomes more regular during the evolution and is closer to the real motion, which for this frame goes diagonally from top left to bottom right.

Acknowledgments

We thank U. Clarenz, G. Dziuk, K. Mikula, and J. Weickert for inspiring discussions and helpful comments on the topic. For the ultrasound data show in Figs. 2 and 4 we thank TomTec Imaging Systems and C. Lamberti from DEIS, Bologna University. Furthermore we acknowledge S. Oswald from Sheffield University, who performed the fresh and salt water experiments shown in Fig. 6.

APPENDIX A. THE CASE OF A GENERAL MOTION

Let us in this additional section consider a motion described by an arbitrary velocity field. As described in Section 2.2, given solely the image sequence resulting from the motion, it turns out to be an ill-posed problem to extract the original velocity. However, let us describe, which information according tangential movements can be extracted and what the apparent velocity really describes.

The simultaneous motion of all points in Ω obeying the velocity field v induces a flux $\Theta_\tau(s_0, x_0) = x(s_0 + \tau)$, with $x(s_0) = x_0$ and $\dot{x}(s) = v(s, x(s))$. We can regard this flux also as the deformation of the image at time $s_0 + \tau$ with respect to the initial state at time s_0. The levelset normals at time $s_0 + \tau$ transform under the flux. This transformation is controlled by the inverse transpose of the Jacobian of the flux:

$$N(s_0 + \tau, x(s_0 + \tau)) = |(\nabla\Theta_\tau)^{-T} N|^{-1} (\nabla\Theta_\tau)^{-T} N(s_0, x_0), \tag{10}$$

We can express this transformation in terms of v using a development in τ up to first order. Expanding the Jacobian of the flux we obtain

$$\nabla\Theta_\tau = \mathrm{Id} + \tau\nabla v + o(\tau^2), \qquad \Rightarrow \qquad (\nabla\Theta_\tau)^{-T} = \mathrm{Id} - \tau(\nabla v)^T + o(\tau^2).$$

The differentiation of the identity (10) with respect to τ at $\tau = 0$ will lead us further. Recalling that for some vector valued function $A(\tau)$

$$\frac{\partial}{\partial\tau} |A(\tau)|\Big|_{\tau=0} = \frac{A(0)}{|A(0)|} \cdot \left(\frac{\partial}{\partial\tau} A(\tau)\Big|_{\tau=0} \right)$$

holds, we can compute from (10) the relation (omitting the arguments for the clearness of the presentation)

$$0 = \Big[\partial_\tau N(s_0 + \tau, x(s_0 + \tau)) + \partial_\tau |(\nabla\Theta_\tau)^{-T} N| N - \partial_\tau (\nabla\Theta_\tau)^{-T} N\Big] \Big|_{\tau=0} = \partial_s N + \nabla N v - (\nabla v)^T N \cdot N N + (\nabla v)^T N.$$

Using $\nabla N = S$ and $\nabla(v \cdot N) = (\nabla v)^T N + (\nabla N)^T v = (\nabla v)^T N + S v$, we can rewrite the latter result as

$$0 = [\partial_s N + S v] + (\mathrm{Id} - N \otimes N) \left(\nabla(v \cdot N) - S v \right). \tag{11}$$

This is the (implicit) expression to compute the tangential velocity for arbitrary motion. Whenever the last term vanishes, i.e. v solves the differential equation $(\mathrm{Id} - N \otimes N)\left(\nabla(v \cdot N) - Sv\right) = 0$, we have a coincidence of apparent velocity and true velocity $v_{\mathrm{app}} = v$, since the remaining left term $0 = [\partial_s N + Sv]$ exactly defines v_{app}. Unfortunately there is not more information contained in (11) because it again is an equation on the tangent bundle. And since the equation itself is implicit in v it is not possible to obtain any measure for the difference between v_{app} and v.

A last observation concerning the case $v = v_{\mathrm{app}}$ is the following: Substituting v_{app} into (11) and using the fact that $(\mathrm{Id} - N \otimes N)S = S$ one obtains

$$(v_{\mathrm{app}})_{\mathrm{tg}} = S^{-1} (\mathrm{Id} - N \otimes N) \nabla((v_{\mathrm{app}})_{\mathrm{n}}),$$

because the term in brackets vanished for v_{app} in (11). This means, the tangential component of the apparent velocity is the variation of the normal velocity weighted on the tangent space according to the shape of the levelset.

REFERENCES

1. L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel. Axioms and fundamental equations of image processing. *Arch. Ration. Mech. Anal.*, 123(3):199–257, 1993.
2. L. Alvarez, J. Weickert, and J. Sánchez. A scale–space approach to nonlocal optical flow calculations. In *Scale-Space 1999, Corfu, Greece, Sept. 1999*, Lecture Notes in Computer Science; 1682, pages 235–246. Springer, 1999.
3. S. B. Angenent and M. E. Gurtin. Multiphase thermomechanics with interfacial structure 2, evolution of an isothermal interface. *Arch. Rational Mech. Anal.*, 108:323–391, 1989.
4. J. Becker and M. Rumpf. Visualization of time-dependent velocity fields by texture transport. In *Proceedings of the Eurographics Scientific Visualization Workshop ’98*. Springer, 1998.
5. D. Bürkle, T. Preußer, and M. Rumpf. Transport and anisotropic diffusion in time-dependent flow visualization. In *Proceedings Visualization ’01*, 2001.
6. B. Cabral and L. Leedom. Imaging vector fields using line integral convolution. In J. T. Kajiya, editor, *Computer Graphics (SIGGRAPH ’93 Proceedings)*, volume 27, pages 263–272, Aug. 1993.
7. G. E. Christensen, S. C. Joshi, and M. I. Miller. Volumetric transformations of brain anatomy. *IEEE Trans. Medical Imaging*, 16(6):864–877, 1997.
8. U. Clarenz, U. Diewald, and M. Rumpf. Nonlinear anisotropic diffusion in surface processing. In *Proc. Visualization 2000*, pages 397–405, 2000.

9. C. A. Davatzikos, R. N. Bryan, and J. L. Prince. Image registration based on boundary mapping. *IEEE Trans. Medical Imaging*, 15(1):112–115, 1996.
10. K. Deckelnick and G. Dziuk. Discrete anisotropic curvature flow of graphs. *Mathematical Modelling and Numerical Analysis - RAIRO*, 33(6):1203–1222, 1999.
11. R. Deriche, P. Kornprobst, and G. Aubert. Optical–flow estimation while preserving its discontinuities: A variational approach. In *Proc. Second Asian Conf. Comp. Vis. (ACCV '95, Singapore)*, volume 2, pages 290–295, 1995.
12. U. Diewald, T. Preußer, and M. Rumpf. Anisotropic diffusion in vector field visualization on euclidean domains and surfaces. *Trans. Vis. and Comp. Graphics*, 6(2):139–149, 2000.
13. M. P. do Carmo. *Riemannian Geometry*. Birkhäuser, Boston–Basel–Berlin, 1993.
14. L. Evans and J. Spruck. Motion of level sets by mean curvature I. *J. Diff. Geom.*, 33(3):635–681, 1991.
15. U. Grenander and M. I. Miller. Computational anatomy: An emerging discipline. *Quarterly Appl. Math. LVI*, pages 617–694, 1998.
16. F. Guichard. A morphological, affine, and galilean invariant scale–space for movies. *IEEE Transactions on Image Processing*, 7(3):444–456, 1998.
17. G. Huisken. The volume preserving mean curvature flow. *J. Reine Angew. Math.*, 382:35–48, 1987.
18. V. Interrante and C. Grosch. Stragegies for effectively visualizing 3d flow with volume lic. In *Proceedings Visualization '97*, pages 285–292, 1997.
19. S. C. Joshi and M. I. Miller. Landmark matching via large deformation diffeomorphisms. *IEEE Trans. Medical Imaging*, 9(8):1357–1370, 2000.
20. F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multi–modal volume registration by maximization of mutual information. *IEEE Trans. Medical Imaging*, 16(7):187–198, 1997.
21. N. Max, R. Crawfis, and C. Grant. Visualizing 3D Velocity Fields Near Contour Surface. In *Proceedings of IEEE Visualization '94*, pages 248–254, 1994.
22. K. Mikula and N. Ramarosy. Semi–implicit finite volume scheme for solving nonlinear diffusion equations in image processing. *Numerische Mathematik*, 2001.
23. H. H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from images sequences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8:565–593, 1986.
24. P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12:629–639, 1990.
25. T. Preußer and M. Rumpf. A level set method for anisotropic diffusion in 3D image processing. *SIAM J. Appl. Math.*, 62(5):1772–1793, 2001.
26. T. Preußer and M. Rumpf. Morphological image sequence processing. *Computing*, 2003, to appear.
27. E. Radmoser, O. Scherzer, and J. Weickert. Scale–space properties of regularization methods. In *Scale-Space '99, Corfu, Greece, Sept. 1999*, Lecture Notes in Computer Science; 1682, pages 211–220. Springer, 1999.
28. G. Sapiro. Vector (self) snakes: A geometric framework for color, texture, and multiscale image segmentation. In *Proc. IEEE International Conference on Image Processing, Lausanne*, September 1996.
29. G. Sapiro and A. Tannenbaum. Affine invariant scale-space. *International Journal of Computer Vision*, 11(1):25–44, 1993.
30. J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
31. H.-W. Shen and D. L. Kao. Uflic: A line integral convolution algorithm for visualizing unsteady flows. In *Proceedings Visualization '97*, pages 317–322, 1997.
32. D. Stalling and H.-C. Hege. Fast and resolution independent line integral convolution. In *SIGGRAPH 95 Conference Proceedings*, pages 249–256. ACM SIGGRAPH, Addison Wesley, Aug. 1995.
33. J. E. Taylor, J. W. Cahn, and C. A. Handwerker. Geometric models of crystal growth. *Acta metall. mater.*, 40:1443–1474, 1992.
34. J. P. Thirion. Image matching as a diffusion process: An analogy with maxwell's demon. *Medical Imag. Analysis 2*, pages 243–260, 1998.
35. J. J. van Wijk. Spot noise-texture synthesis for data visualization. In T. W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 309–318, July 1991.
36. J. J. van Wijk. Image based flow visualization. In *ACM Transactions on Graphics, Special Issue, Proc. ACM Siggraph 02*, 2002.
37. J. Weickert. *Anisotropic diffusion in image processing*. Teubner, 1998.