A NEW NUMERICAL METHOD FOR LEVEL SET MOTION IN NORMAL DIRECTION USED IN OPTICAL FLOW ESTIMATION

Preprint · January 2019

CITATIONS
0

READS
7

2 authors, including:

Peter Frolkovič
Slovak University of Technology in Bratislava
56 PUBLICATIONS **541** CITATIONS

SEE PROFILE

Manuscript submitted to
AIMS' Journals
Volume **X**, Number **0X**, XX **X**

doi:xx.xxxx/xx.xx.xx.xx

pp. **X–XX**

A NEW NUMERICAL METHOD FOR LEVEL SET MOTION IN NORMAL DIRECTION USED IN OPTICAL FLOW ESTIMATION

Peter Frolkovič* and Viera Kleinová

Faculty of Civil Engineering, Slovak University of Technology
Department of Mathematics and Descriptive Geometry
Radlinského 11, 810 05 Bratislava, Slovak Republic

(Communicated by the associate editor name)

Abstract. We present a new numerical method for the solution of level set advection equation describing a motion in normal direction for which the speed is given by the sign function of the difference of two given functions. Taking one function as the initial condition, the solution evolves towards the second given function. One of possible applications is an optical flow estimation to find a deformation between two images in a video sequence. The new numerical method is based on a bilinear interpolation of discrete values as used for the representation of images. Under natural assumptions it insures a monotone evolution of the numerical solution towards the given function, and it handles properly the discontinuity in the speed due to the dependence on the sign function. To find a deformation between two functions (or images) the backward tracking of characteristics is used. Two numerical experiments are presented, one with an exact solution to show an experimental order of convergence and one based on two images of lungs to illustrate a possible application of the method for the optical flow estimation.

1. **Introduction.** One of the most common mathematical tool in level set methods is a solution of nonlinear advection equation that describes a motion of evolving function in normal direction to its level sets. Very often the level set method is used to track a position of a dynamic interface (e.g., a curve in 2D case) that is defined only implicitly as a zero level set of the evolving function. For a review of diverse applications of level set methods we refer to [12, 10, 7].

The main motivation of our study is a possible application of level set methods for so-called optical flow estimation between two greyscale images in a video sequence when one searches a deformation of one image, a source, into another one, a target. We refer to [8, 1, 13, 2, 5, 3] for more detailed descriptions of this problem used in image processing, we restrict here on a particular task of solving the nonlinear advection equation of which the solution evolves from the given source function towards the given target function by the level set motion in normal direction [1, 13].

We derive a new numerical scheme that handles appropriately a discontinuity of the speed in the model and that uses a form of numerical solution typical for

2010 *Mathematics Subject Classification.* Primary: 65D18, 65M25; Secondary: 35L60.

Key words and phrases. optical flow estimation, level set equation, motion in normal direction.

Work supported by grants VEGA 1/0728/15, APVV-15-0522 and APVV-16-0431. The authors are grateful for a support of company Tatramed in Bratislava, Slovakia.

the representation of images. In particular, the method shall approximate the solution using the bilinear interpolation, and it shall give under natural assumptions a monotone evolution of the numerical solution towards the given target function even when the speed of evolution allows a change of its sign.

For that purpose we extend a so-called Rouy-Tourin scheme [11, 12] for the numerical solution of nonlinear advection equation using an idea of so-called Corner Transport Upwind scheme that is described for linear advection equation in [4, 9, 6]. The Rouy-Tourin scheme is based on the linear interpolation while the proposed scheme uses the bilinear interpolation. By a careful definition of time steps we insure that the new scheme gives a monotone evolution of the numerical solution towards the target function.

The proposed scheme is given not only in an Eulerian form to define the values of numerical solution in fixed points of rectangular grids, but also in a semi-Lagrangian form to obtain the numerical solution by tracking backward the points along characteristic curves. If the source and target functions differs significantly one has to apply the scheme several times up to some stopping criteria are fulfilled. For such cases we define a modification of the backward tracking method that does not require to store intermediate results of all time steps.

The paper is organized as follows. In Section 2 we describe the model of the motion in the normal direction for the level sets of evolving function towards the level sets of target function. In Section 3 we propose the new numerical method to solve the model approximately including the backward tracking of characteristic curves. In Section 4 we present two numerical experiments, one with a known exact solution and one with an illustrative application of the optical flow estimation using real biological images of lungs.

2. Level set motion from a source function towards a target function.

Let $F = F(x)$ and $G = G(x)$ be given functions defined on a domain $\Omega \subset \mathbb{R}^2$. The functions F and G are called the source function and the target function, respectively. Our aim is to evolve the source F by a motion of level sets in normal direction towards the target G.

For that purpose we search for a function $f = f(x,t)$ fulfilling the level set equation

$$\partial_t f = \operatorname{sgn}(G-f)|\nabla f|, \quad f(x,0) = F(x)\,, \quad x \in \Omega\,,\ t \geq 0\,, \tag{1}$$

where sgn is the standard sign function. The equation (1) can be equivalently formulated in the form of nonlinear advection equation

$$\partial_t f + \vec{u}\cdot\nabla f = 0, \quad f(x,0) = F(x)\,, \quad x \in \Omega\,,\ t \geq 0\,, \tag{2}$$

with

$$\vec{u} = \begin{cases} -\operatorname{sgn}(G-f)\frac{\nabla f}{|\nabla f|} & |\nabla f| \neq 0 \\ \vec{0} & |\nabla f| = 0\,. \end{cases} \tag{3}$$

The vector field $\vec{u}$, if nonzero, represents the normal vectors to the level sets of f with the orientation given by the sign of $G-f$.

The equation (1), respectively (2), prescribes the movement of level sets of the function f in normal direction [12, 10] with the speed changing abruptly from the values 1 or -1 to 0 if $f = G$ or $\nabla f = \vec{0}$ that must be treated carefully in numerical methods.

From (1) one obtains that the difference $|G(x) - f(x,t)|$ for a fixed x is monotonically decreasing in time if $f(x,t) \neq G(x)$ and $\nabla f(x,t) \neq \vec{0}$ as

$$\frac{1}{2}\frac{d}{dt}\left(G(x) - f(x,t)\right)^2 = \tag{4}$$
$$-(G(x) - f(x,t))\partial_t f(x,t) = -|G(x) - f(x,t)||\nabla f(x,t)| < 0\,.$$

Furthermore, if for some $T \geq 0$ one has for all $x \in \Omega$ that $f(x,T) = G(x)$ or $|\nabla f(x,T)| = 0$ then $f(x,t) \equiv f(x,T)$ for $t \geq T$. In this sense we say that the function f is evolving from the source function F towards the target function G. In general one may try to avoid $|\nabla f| = 0$ in (1) by using some pre-processing of this term, see, e.g., [13], that we do not consider here.

Concerning boundary conditions for the equations (1) or (2) we set the normal component of ∇f to zero at the boundary $\partial\Omega$ of Ω.

In the section on numerical methods we derive a new numerical scheme to solve (1), respectively (2), for which the obtained numerical solution will fulfill the property (4) in a discrete form. This scheme gives the approximate solution in fixed positions of grid points that corresponds to an Eulerian type of methods [9]. Next we consider the problem from a view of Lagrangian type of methods.

Once the solution f of (2) is known one can compute the characteristic curves $X = X(t)$ generated by $\vec{u}$ in (3) by solving backward in time the ordinary differential equations

$$\dot{X} = \vec{u}(X,t), \quad X(\tilde{t}) = x \tag{5}$$

for some $x \in \Omega$ and $\tilde{t} > 0$, so we may write $X = X(t) = X(x,\tilde{t};t)$ and $t \in [0,\tilde{t}]$. The value $X(x,\tilde{t};t)$ denotes the position X at time t of the characteristic curve for which the position at time $\tilde{t}$ is x.

For the solution $f(x,t)$ of (2) the time derivative of $f(X(t),t)$ vanishes,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}f(X(t),t) &= \partial_t f(X(t),t) + \dot{X}(t)\cdot\nabla f(X(t),t)\\ &= \partial_t f(X(t),t) + \vec{u}(X(t),t)\cdot\nabla f(X(t),t) = 0\,.\end{aligned} \tag{6}$$

Consequently, the function f is constant along the characteristic curves, e.g., $f(X(t),t) \equiv f(x,\tilde{t})$, $t \in [0,\tilde{t}]$. Let us now suppose that the solution f of (1) is available for some interval $[0,T]$ with large enough time $T > 0$. If one tracks backward the characteristic curves in (5) for $x \in \Omega$ from $t = \tilde{t} = T$ to $t = 0$ we obtain

$$f(x,T) = f(X(x,T;T),T) = f(X(x,T;t),t) = f(X(x,T;0),0) = F(X(x,T;0))\,.$$

Consequently, we can define a deformation

$$\vec{D} = \vec{D}(x) = x - X(x,T;0) \tag{7}$$

for which we can distinguish two situations.

Firstly, if $f(x,T) = G(x)$ then

$$G(x) = F(x - \vec{D}(x))\,. \tag{8}$$

If this property is valid for all $x \in \Omega$, the deformation $\vec{D}$ can be viewed as the optical flow estimation [8].

Secondly, if $f(x,T) \neq G(x)$ then $G(x) \neq F(x - \vec{D}(x))$. This can happen in practice either if T is not large enough or $|\nabla f(x,t)| \equiv 0$ for $t \geq T$. Nevertheless, under analogous assumptions as used for (4), one has

$$\frac{1}{2}\frac{d}{dt}\left(G(x) - F(x,t;0)\right)^2 < 0\,.$$

In the next section we propose an approximation of the deformation $\vec{D}$ in (7) using a numerical method to approximate the characteristics $X(x,t;0)$.

3. **Numerical methods.** We suppose that the functions F and G in (1) are represented by their discrete values $F_{ij} = F(x_{ij})$ and $G_{ij} = G(x_{ij})$ in the points $x_{ij} := (ih, jh)$ such that $i = 0,\dots,I$, $j = 0,\dots,J$, and $h > 0$ is a given constant. The values $F(x)$ for $x \in ((i-1)h, ih) \times ((j-1)h, jh)$, $i > 0$, $j > 0$ are obtained by the standard bilinear interpolation of related four discrete values F_{ij}, F_{i-1j}, F_{ij-1}, F_{i-1j-1}.

First, we consider a numerical approximation of the level set equation (1). Let τ be a chosen fixed time step and $t^n = n\tau$. We suppose that for some $n \geq 0$ the following values are available

$$f_{ij}^n \approx f(x_{ij}, t^n) = F(X(x_{ij}, t^n; 0)) \tag{9}$$

with the case $n = 0$ defined by $f_{ij}^0 = F_{ij}$. Furthermore, we initialize

$$s_{ij} = \operatorname{sgn}(G_{ij} - F_{ij}) \tag{10}$$

and we suppose that $\operatorname{sgn}(G_{ij} - f_{ij}^n) = \operatorname{sgn}(G_{ij} - F_{ij})$. Note that if $f_{ij}^n = G_{ij}$ we use $f_{ij}^{n+k} \equiv f_{ij}^n$ for $k = 1, 2, \dots$ that can be obtained by setting $s_{ij} = 0$.

Firstly we quote Rouy-Tourin numerical scheme [11, 12] to approximate (1),

$$_{RT}f_{ij}^{n+1} = f_{ij}^n + \tau s_{ij} |\nabla f_{ij}^n| \,, \tag{11}$$

where the approximation of $\nabla f_{ij}^n = (\partial_{x_1} f_{ij}^n, \partial_{x_2} f_{ij}^n) \approx \nabla f(x_{ij}, t^n)$ is given by

$$h\partial_{x_1} f_{ij}^n = \begin{cases} f_{ij}^n - f_{i-1j}^n & f_{i-1j}^n = \operatorname{ext}\{f_{i-1j}^n, f_{ij}^n, f_{i+1j}^n\} \\ f_{i+1j}^n - f_{ij}^n & f_{i+1j}^n = \operatorname{ext}\{f_{i-1j}^n, f_{ij}^n, f_{i+1j}^n\} \\ 0 & f_{ij}^n = \operatorname{ext}\{f_{i-1j}^n, f_{ij}^n, f_{i+1j}^n\} \end{cases} \tag{12}$$

$$h\partial_{x_2} f_{ij}^n = \begin{cases} f_{ij}^n - f_{ij-1}^n & f_{ij-1}^n = \operatorname{ext}\{f_{ij-1}^n, f_{ij}^n, f_{ij+1}^n\} \\ f_{ij+1}^n - f_{ij}^n & f_{ij+1}^n = \operatorname{ext}\{f_{ij-1}^n, f_{ij}^n, f_{ij+1}^n\} \\ 0 & f_{ij}^n = \operatorname{ext}\{f_{ij-1}^n, f_{ij}^n, f_{ij+1}^n\} \end{cases} \tag{13}$$

The notation ext denotes the extreme values among the three discrete values of f in (12) or (13) with the particular choice depending on the sign of s_{ij}, namely

$$\operatorname{ext} = \begin{cases} \min & s_{ij} < 0 \\ \max & s_{ij} > 0 \,. \end{cases}$$

The definitions (12) and (13) are modified for boundary nodes with prescribed zero normal component of ∇f simply by skipping the non-existing discrete values f_{ij}^n with $i = -1$, $i = I+1$, $j = -1$, or $j = J+1$ in the argument of ext.

Next we rewrite the scheme (11) in an equivalent form that help us to use it for our purpose. To do so let $k, l \in \{-1, 1, 0\}$ be such that the value f_{i+kj}^n is the extreme value chosen in (12), and, analogously, the value f_{ij+l}^n is chosen in (13). Consequently one has that $h|\partial_{x_1} f_{ij}^n| = |f_{i+kj}^n - f_{ij}^n| = s_{ij}(f_{i+kj}^n - f_{ij}^n)$ and $h|\partial_{x_2} f_{ij}^n| = |f_{ij+l}^n - f_{ij}^n| = s_{ij}(f_{ij+l}^n - f_{ij}^n)$.

If $|\nabla f_{ij}^n| = 0$ then, clearly, $f_{ij}^{n+1} = f_{ij}^n$. In what follows we assume that $|\nabla f_{ij}^n| \neq 0$, and we define "directional Courant numbers" [9, 6] for the fixed time step τ,

$$\mathcal{C}_{ij}^n = \frac{\tau}{h} \frac{|\partial_{x_1} f_{ij}^n|}{|\nabla f_{ij}^n|} \,, \quad \mathcal{D}_{ij}^n = \frac{\tau}{h} \frac{|\partial_{x_2} f_{ij}^n|}{|\nabla f_{ij}^n|} \,, \tag{14}$$

and the scheme (11) can be written in the form

$$ {}_{RT}f_{ij}^{n+1} = (1-\mathcal{C}_{ij}^n-\mathcal{D}_{ij}^n)f_{ij}^n + \mathcal{C}_{ij}^n f_{i+kj}^n + \mathcal{D}_{ij}^n f_{ij+l}^n \,. \tag{15} $$

Clearly, the scheme (15) defines the value ${}_{RT}f_{ij}^{n+1}$ as a convex combination of three values if

$$ \mathcal{C}_{ij}^n + \mathcal{D}_{ij}^n \le 1 \,. \tag{16} $$

The sufficient condition for (16) is that $\tau \le h/\sqrt{2}$ in (14).

In what follows we modify the scheme (15) using an idea of so called Corner Transport Upwind scheme [4, 9, 6]. Opposite to the scheme (15) that uses the linear interpolation of three values, the following scheme will use the bilinear interpolation involving also the corner value f_{i+kj+l}^n, namely

$$ \begin{aligned} {}_{CTU}f_{ij}^{n+1} = (1-\mathcal{C}_{ij}^n-\mathcal{D}_{ij}^n)f_{ij}^n + \mathcal{C}_{ij}^n f_{i+kj}^n + \mathcal{D}_{ij}^n f_{ij+l}^n + \\ + \mathcal{C}_{ij}^n\mathcal{D}_{ij}^n\left(f_{ij}^n - f_{i+kj}^n - f_{ij+l}^n + f_{i+kj+l}^n\right) \,. \end{aligned} \tag{17} $$

The right hand side of (17) is a convex combination of four values if

$$ \max\{\mathcal{C}_{ij}^n, \mathcal{D}_{ij}^n\} \le 1 \tag{18} $$

that is less restrictive than (16). A sufficient condition to obtain (18) is that $\tau \le h$ in (14).

Next we formulate the scheme (17) using a quadratic function of τ on the right hand side, namely

$$ {}_{CTU}f_{ij}^{n+1} = \tilde{f}_{ij}^{n+1}(\tau)\,, \quad \tilde{f}_{ij}^{n+1}(\tau) := f_{ij}^n + \tau s_{ij}|\nabla f_{ij}^n| + \tau^2\alpha_{ij}^n \tag{19} $$

and

$$ \alpha_{ij}^n := \frac{\left(f_{ij}^n - f_{i+kj}^n - f_{ij+l}^n + f_{i+kj+l}^n\right)|\partial_{x_1}f_{ij}^n\,\partial_{x_2}f_{ij}^n|}{h^2|\nabla f_{ij}^n|^2} \,. \tag{20} $$

Note that if $f_{ij}^n \neq G_{ij}$ then analogously to (4) one has

$$ \frac{d}{d\tau}\frac{1}{2}(G_{ij}-\tilde{f}_{ij}^{n+1}(\tau))^2\Big|_{\tau=0} = -|G_{ij}-f_{ij}^n||\nabla f_{ij}^n| < 0 \,. $$

To finalize our numerical schemes we have to suggest proper values of the time step τ to be used in (19). Our aim is to use the maximal value $\tau = h$ to speed up the evolution of the values $\tilde{f}_{ij}^{n+1}(\tau)$ towards the values G_{ij}. Nevertheless there are two cases when the scheme (19) has to be modified for such time step. We define the modifications using a variable choice of τ in (19), say $\tau = \tau_{ij}^n$, but we discuss also equivalent modifications of (19) for which $\tau = h$.

We distinguish two cases. In the case 1 we check if $\operatorname{sgn}(G_{ij}-\tilde{f}_{ij}^{n+1}(h)) \neq \operatorname{sgn}(G_{ij}-f_{ij}^n) = s_{ij}$. If this happens there exists $\tau < h$, say τ_{ij}^n, such that $\tilde{f}_{ij}^{n+1}(\tau_{ij}^n) = G_{ij}$ that is our desired property. Consequently, we solve a quadratic equation for τ_{ij}^n (if $\alpha_{ij}^n \neq 0$)

$$ f_{ij}^n - G_{ij} + \tau_{ij}^n s_{ij}|\nabla f_{ij}^n| + (\tau_{ij}^n)^2\alpha_{ij}^n = 0 \,. \tag{21} $$

The scheme (19), respectively (17), shall be considered only for $0 < \tau \le \tau_{ij}^n$, and one shall consider ${}_{CTU}f_{ij}^{n+1} \equiv G_{ij}$ for $\tau_{ij}^n \le \tau \le h$. We use simply (19) with the variable time step $\tau = \tau_{ij}^n$.

In the case 2 we want to avoid that for some $\tau < h$, say again τ_{ij}^n, the derivative $\frac{d}{d\tau}\tilde{f}_{ij}^{n+1}(\tau_{ij}^n)$ vanishes, so the difference $|G_{ij}-\tilde{f}_{ij}^{n+1}(\tau)|$ becomes increasing for $\tau >$

τ_{ij}^n. Clearly, this property is not desired, therefore we search for τ_{ij}^n solving a linear equation (if $\alpha_{ij}^n \neq 0$)

$$s_{ij}|\nabla f_{ij}^n| + 2\tau_{ij}^n \alpha_{ij}^n = 0 \tag{22}$$

and we again use (19) with the variable time step $\tau = \tau_{ij}^n$.

Another approach is to modify the scheme (19) using an idea of "limiters" [9] by "limiting" the value α_{ij}^n towards zero when the scheme (19) is approaching the form of scheme (11). The limited version of (19) with the time step $\tau = h$ shall give then the same value as the unlimited form (19) with $\tau = \tau_{ij}^n$. Again for a simplicity we prefer the second form with the variable time step.

We are now ready to define the time steps τ_{ij}^n in (19) according to (21) and (22) in details. It is enough to define it only if $|\nabla f_{ij}^n| \neq 0$ and $f_{ij}^n \neq G_{ij}$.

Firstly, if $\alpha_{ij}^n = 0$ then $\tilde{f}_{ij}^{n+1}(\tau)$ becomes a linear function. Consequently, to control the case 1 we require

$$\tau_{ij}^n = \min\{h, \frac{|G_{ij} - f_{ij}^n|}{|\nabla f_{ij}^n|}\}\,. \tag{23}$$

Let us now consider that $\alpha_{ij}^n \neq 0$. We denote the discriminant of quadratic equation (21) by

$$D = |\nabla f_{ij}^n|^2 + 4s_{ij}|G_{ij} - f_{ij}^n|\alpha_{ij}^n\,. \tag{24}$$

If $D < 0$ then there exists no τ such that $\tilde{f}_{ij}^{n+1}(\tau) = G_{ij}$, therefore we need to control only the case 2 by requiring

$$\tau_{ij}^n = \min\left\{h, -\frac{s_{ij}|\nabla f_{ij}^n|}{2\alpha_{ij}^n}\right\}\,. \tag{25}$$

Note that if $D < 0$ then $\mathrm{sgn}(\alpha_{ij}^n) = -s_{ij}$, so one has that $\tau_{ij}^n > 0$ in (25).

If $D > 0$ then there exists at least one value of τ such that $\tilde{f}_{ij}^{n+1}(\tau) = G_{ij}$. Consequently, we control the case 1 by

$$\tau_{ij}^n = \min\{h, \frac{s_{ij}\left(\sqrt{D} - |\nabla f_{ij}^n|\right)}{2\alpha_{ij}^n}\}\,. \tag{26}$$

Note that if $s_{ij}\alpha_{ij}^n < 0$ then $\sqrt{D} < |\nabla f_{ij}^n|$ and if $s_{ij}\alpha_{ij}^n > 0$ then $\sqrt{D} > |\nabla f_{ij}^n|$, so one has that $\tau_{ij}^n > 0$ in (26).

We can now summarize that depending on the value of α_{ij}^n in (20) and D in (24) one can define the value of τ_{ij}^n by (23), (25), or (26) that can be used in (19) to define the values $\tilde{f}_{ij}^{n+1}(\tau_{ij}^n)$.

Defining the directional Courant numbers with the variable time steps τ_{ij}^n

$$c_{ij}^n = \frac{\tau_{ij}^n}{h}\frac{|\partial_{x_1} f_{ij}^n|}{|\nabla f_{ij}^n|}\,, \quad d_{ij}^n = \frac{\tau_{ij}^n}{h}\frac{|\partial_{x_2} f_{ij}^n|}{|\nabla f_{ij}^n|}\,, \tag{27}$$

the scheme (17) turns to our final form in the Eulerian form

$$\begin{aligned} {}_E f_{ij}^{n+1} = (1 - c_{ij}^n - d_{ij}^n) f_{ij}^n + c_{ij}^n f_{i+kj}^n + d_{ij}^n f_{ij+l}^n + \\ c_{ij}^n d_{ij}^n (f_{ij}^n - f_{i+kj}^n - f_{ij+l}^n + f_{i+kj+l}^n)\,. \end{aligned} \tag{28}$$

The values ${}_E f_{ij}^{n+1} \approx f(x_i, y_j, t^n)$ in (28) approximates the evolution of f from F towards G in the Eulerian framework. Such framework is used for a specific application of optical flow in [1].

Next we introduce it also in the semi-Lagrangian framework by defining $X_{ij}^{n+1} \approx X(x_{ij}, t^{n+1}; 0)$. Note that a standard approach to do it is to compute a numerical solution of ordinary differential equations (5) for which the approximations of $\vec{u}(x, t^m)$ for all previous time steps $m = 0, 1, \ldots, n$ must be available. We propose an alternative approach.

To do so we suppose that some approximation $X_{ij}^n \approx X(x_{ij}, t^n; 0)$ is available. Clearly, $X_{ij}^0 = x_{ij}$. To define X_{ij}^{n+1} we use the bilinear interpolation analogously to (28)

$$X_{ij}^{n+1} = (1 - c_{ij}^n - d_{ij}^n)X_{ij}^n + c_{ij}^n X_{i+kj}^n + d_{ij}^n X_{ij+l}^n + c_{ij}^n d_{ij}^n (X_{ij}^n - X_{i+kj}^n - X_{ij+l}^n + X_{i+kj+l}^n)\,. \tag{29}$$

Finally, to be consistent with our aim to determine an approximation of (8) we define

$${}_L f_{ij}^{n+1} = F(X_{ij}^{n+1})\,. \tag{30}$$

Note that ${}_L f_{ij}^n = {}_E f_{ij}^n$ for $n = 1$, but this property is lost in general for $n > 1$.

The method (30) can be finished at some n when for all $i = 0, 1, \ldots, I$ and $j = 0, 1, \ldots, J$ one has either $|G_{ij} - f_{ij}^n| < \epsilon_1$ or $|\nabla f_{ij}^n| < \epsilon_2$ or $n = N$. In such case the approximation of deformation $\vec{D}$ in (7) in the grid points can be defined by

$$\vec{D}_{ij} = x_{ij} - X_{ij}^n\,. \tag{31}$$

We now summarize the method written in a pseudo code description:

- take input F_{ij}, G_{ij}, ϵ_1, ϵ_2, N;
- initialize $s_{ij} = \mathrm{sgn}(G_{ij} - F_{ij})$, $X_{ij}^0 = x_{ij}$, $n = 0$, notfinished=1;
- while (notfinished & $n \leq N$) do:
  - notfinished = 0;
  - for $i = 0, 1, \ldots, I$ and $j = 0, 1, \ldots, J$ do:
    - if $s_{ij} = 0$ then continue;
    - compute ∇f_{ij}^n in (12) - (13); if $|\nabla f_{ij}^n| < \epsilon_2$ then continue;
    - notfinished=1;
    - compute α_{ij}^n in (20);
    - if $\alpha_{ij}^n = 0$ then compute τ_{ij}^n in (23);
    - if $\alpha_{ij}^n \neq 0$ then
      1. compute D in (24);
      2. if $D < 0$ then compute τ_{ij}^n in (25) else compute τ_{ij}^n in (26);
    - compute ${}_E f_{ij}^{n+1}$ in (28) using (27) ;
    - compute X_{ij}^{n+1} in (29) and ${}_L f_{ij}^{n+1}$ in (30);
    - if $|G_{ij} - {}_L f_{ij}^{n+1}| < \epsilon_1$ or $|G_{ij} - {}_E f_{ij}^{n+1}| < \epsilon_1$ then set $s_{ij} = 0$;
  - $n = n + 1$;
- for $i = 0, 1, \ldots, I$ and $j = 0, 1, \ldots, J$ compute $\vec{D}_{ij}$ in (31);

In the next section we apply the method to compute (31) for two illustrative examples.

4. Numerical experiments.

In this section we present two examples where the first one shall illustrate the experimental order of convergence of the proposed method and the second one illustrates a possible application of the method for an optical flow estimation of real biological images.

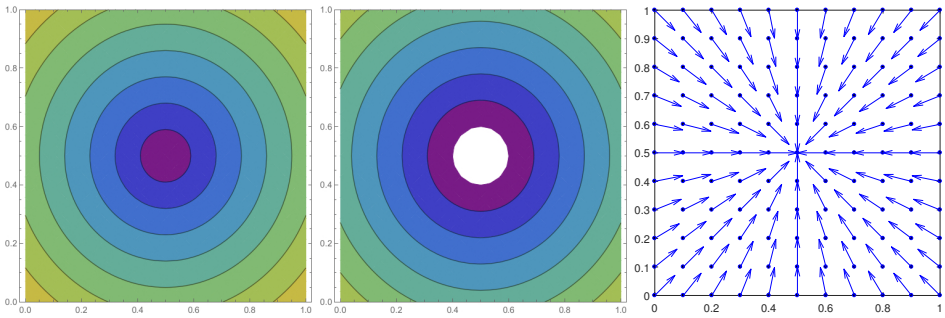


FIGURE 1. The example with exact solution: the function F (left), the function G (middle), and the deformation $-\vec{D}^{ex}$ (right).

4.1. **Example with exact solution.** The domain Ω is a unit square. The functions F and G are defined as follows

$$F(x) = |x - c|\,, \quad G(x) = \max\{0, F(x) - 0.1\}\,, \tag{32}$$

where $c = (0.5, 0.5)$ and $|\cdot|$ denotes the standard Euclidean distance. For a plot of both functions see Figure 1. The exact optical flow is given by the deformation

$$\vec{D}^{ex}(x) = \min\{1, \frac{0.1}{|x - c|}\}\,(x - c)\ . \tag{33}$$

that is plotted in Figure 1 graphically as $-\vec{D}^{ex}$ to show that the value of G at the position x_{ij} (where an arrow starts) equals to the value of F at the position $x_{ij} - \vec{D}^{ex}(x_{ij})$ (where the arrow ends), so $G(x_{ij}) = F(x_{ij} - \vec{D}^{ex}(x_{ij}))$.

The discretization is realized for $I = J = 10, 20, \ldots, 160$ giving the discretization steps $h = 0.1, 0.05, \ldots, 0.00625$. The number of time steps is given by $N = I/10$ and $\epsilon_1 = \epsilon_2 = 0$. The results are presented graphically in Figure 2 where one can observe an improvement of the approximation $\vec{D}_{ij} \approx \vec{D}(x_{ij})$ with finer grids.

Moreover, the discrete L_1-norm is computed for the difference function $E(x) := |G(x) - F(x - \vec{D}(x))|$, i.e.,

$$\|E\| = h^2 \sum_{i,j} E_{ij} = h^2 \sum_{i,j} \left| G_{ij} - F(x_{ij} - \vec{D}_{ij}) \right|\,, \tag{34}$$

and analogously for the first component of the vector $\vec{D}^{ex} - \vec{D}$, where $\vec{D}$ is the bilinear interpolation of obtained numerical values $\vec{D}_{ij}$ (the norm for the second component of $\vec{D}^{ex} - \vec{D}$ is identical). One can observe that the discrete L_1-norms are giving the experimental order of convergence approximately 1 in both cases.

4.2. **Example with images of lungs.** In this example the functions F and G are obtained by the bilinear interpolation of two pictures of the size 263×190 pixels with $h = 1$, see Figure 3 where also the difference picture $|G - F|$ is plotted. The range of values for F and G is between 0 and 192. To plot the difference image $|G - F|$ it is scaled to have the maximal value 256.

We choose a large enough N and $\epsilon_1 = \epsilon_2 = 0$ to show clearly the behavior of the method. The normalized norm

$$e^n = \frac{1}{263 \cdot 190} \sum_{i,j} |G_{ij} - F(X^n_{ij})| \tag{35}$$

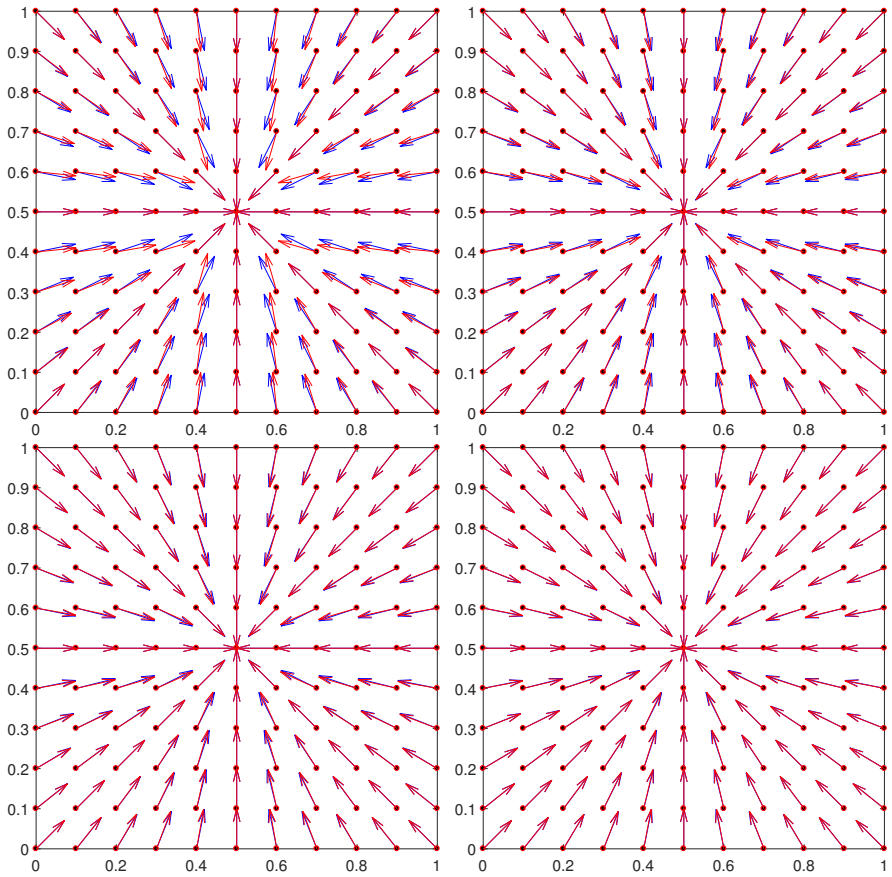


FIGURE 2. Comparison of the exact deformation $\vec{D}^{ex}$ (the blue arrows) with the numerical one $\vec{D}$ (the red arrows) for the grid size $h = 0.1$ (top left), $h = 0.05$ (top right), $h = 0.025$ (bottom left), and $h = 0.0125$ (bottom right).

TABLE 1. The L_1-norms and the experimental rates of convergence for the example with exact solution.

I	N	$\|E\|$	EOC	$\|(\vec{D}^{ex} - \vec{D})_1\|$	EOC
10	1	0.003120	-	0.004433	-
20	2	0.001307	1.2556	0.002379	0.8985
40	4	0.000528	1.3075	0.001259	0.9175
80	8	0.000220	1.2667	0.000659	0.9339
160	16	0.000096	1.1947	0.000339	0.9590

is presented in Figure 4 for $n \leq N = 10$. One can observe a fast decrease of this norm for first time steps.

The image obtained from the interpolation of values $F(x_{ij} - \vec{D}_{ij})$ is plotted in Figure 5 together with the difference image $|G(x) - F(x - \vec{D}(x))|$. The approximation

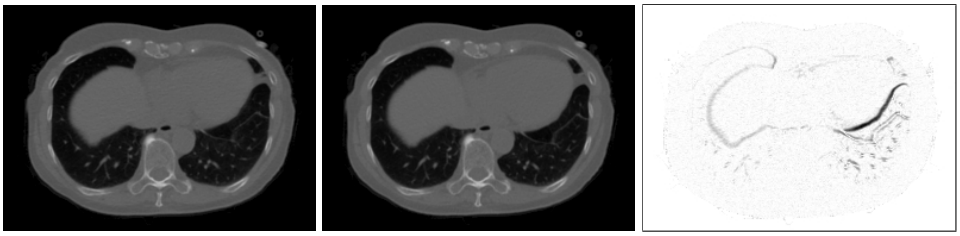
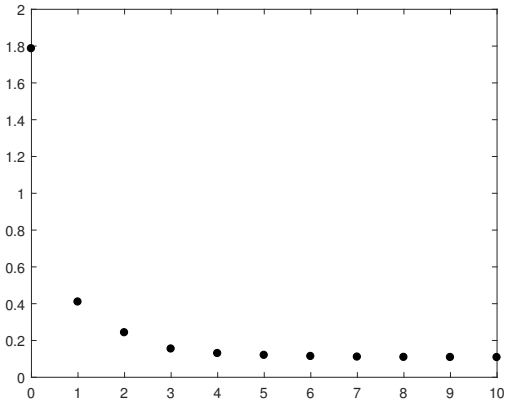


FIGURE 3. The images of lungs scan: the source image F (left), the target image G (middle), the difference image $|G - F|$ (right).



0	1.785792
1	0.409182
2	0.242089
3	0.153696
4	0.129095
5	0.119188
6	0.113130
7	0.109794
8	0.108163
9	0.107613
10	0.107256

FIGURE 4. The plot and the table of the normalized norm e^n in (35).

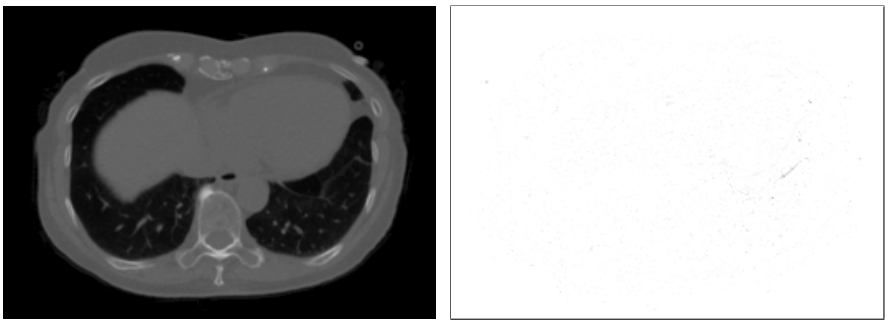


FIGURE 5. The image given by the values $F(x_{ij} - \vec{D}_{ij})$ (left) and the difference image given by the values $|G_{ij} - F(x_{ij} - \vec{D}_{ij})|$ (right).

of deformation $\vec{D}$ is plotted in Figure 6. To make a more clear presentation of this deformation we show it also in Figure 7 where the arrows are plotted only in the points where the values $|G_{ij} - F_{ij}|$ are larger than a chosen value E_{crit}, in this particular case

$$E_{crit} = \frac{1}{15} \max_{i,j} |G_{ij} - F_{ij}| . \tag{36}$$

5. **Conclusions.** We deal with the problem to find a function that evolves from a given source function to a given target function by the flow restricted only to

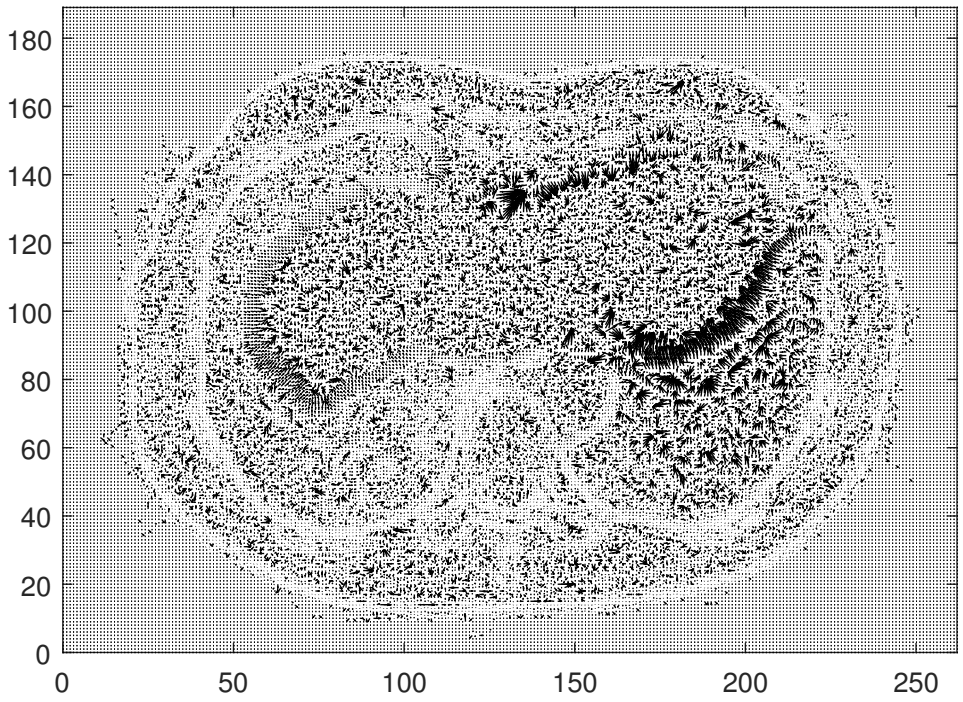


FIGURE 6. The plot of deformation $\vec{D}$ for the example with the images of lungs.

normal direction of level sets. The speed of evolution in the normal direction in the corresponding level set advection equation equals to the sign of the difference of the two given functions, and it changes discontinuously to zero when the evolving function takes the values of target function or when the gradient vanishes. One of possible application of the method is the optical flow estimation that searches a deformation between two images when the given images are represented by the functions defined by the bilinear interpolation of the greyscale values in pixels.

The new presented numerical method for such problems in two-dimensional case has two major novelties. Firstly, opposite to the standard Rouy-Tourin scheme that is based on the linear interpolation defined by three values of the numerical solution, we extend the scheme to use the bilinear interpolation based on four values of the numerical solution. The new method enables larger time steps than the Rouy-Tourin scheme. Secondly, the numerical method insures a monotone evolution of the numerical evolving function under natural assumptions, and the evolution stops sharply when the value of target function is reached locally. Finally, when several time steps shall be used to find the deformation by the backward tracking of characteristic curves we propose its modification that does not require the storage of results for the intermediate steps.

In numerical experiments we give the example with exact solution where the experimental order of convergence gives the expected first order accuracy. Furthermore the example with two images of lungs illustrates a possible usage of the method for optical flow estimation for real biological data.

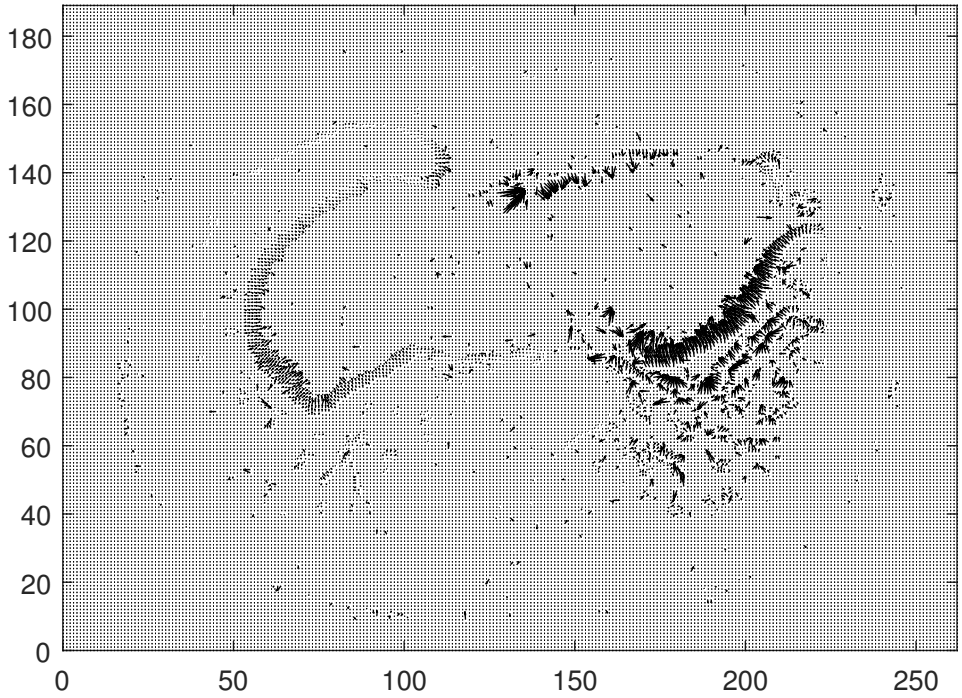


FIGURE 7. The plot of deformation $\vec{D}$ for the example with the images of lungs. Only arrows in the points where $|G_{ij} - F_{ij}| > E_{crit}$ are plotted.

REFERENCES

[1] Marcelo Bertalmío, Guillermo Sapiro, and Gregory Randall. Morphing active contours. *IEEE Trans. PAMI*, 22(7):733–737, 2000.

[2] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/Kanade Meets Horn/Schunck : Combining Local and Global Optic Flow Methods. *Int. J. Comput. Vis.*, 61(3):211–231, 2005.

[3] Martin Burger, Hendrik Dirks, and Carola-Bibiane Sch´onlieb. A variational model for joint motion estimation and image reconstruction. *SIAM Journal on Imaging Sciences*, 11(1):94–128, 2018.

[4] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *J. Comput. Phys.*, 87(1):171–200, 1990.

[5] Valérie Duay, Nawal Houhou, and J-P Thiran. Atlas-based segmentation of medical images locally constrained by level sets. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages II–1286–9, 2005.

[6] Peter Frolkovič and Karol Mikula. Semi-implicit second order schemes for numerical solution of level set advection equation on cartesian grids. *Appl. Math. Comp.*, 329:129–142, 2018.

[7] Frederic Gibou, Ronald Fedkiw, and Stanley Osher. A review of level-set methods and some recent applications. *J. Comput. Phys.*, 353:82–109, 2017.

[8] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[9] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge UP, 2002.

[10] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.

[11] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Num. Anal.*, 29:867–884, 1992.

[12] J. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge UP, 1999.

[13] B. C. Vemuri, J. Ye, Y. Chen, and C. M. Leonard. Image registration via level-set motion: Applications to atlas-based segmentation. *Medical Image Analysis*, 7(1):1–20, 2003.

Received xxxx 20xx; revised xxxx 20xx.

E-mail address: peter.frolkovic@stuba.sk