DISCRETE AND CONTINUOUS
DYNAMICAL SYSTEMS SERIES S

doi:10.3934/dcdss.2020348

AN AUTOMATED SEGMENTATION OF NATURA 2000 HABITATS FROM SENTINEL-2 OPTICAL DATA

Karol Mikula*, Jozef Urbán, Michal Kollár, Martin Ambroz

Department of Mathematics, Slovak University of Technology
Radlinského 11, 810 05 Bratislava, Slovakia
Algoritmy:SK, s.r.o., Šulekova 6, 811 06 Bratislava, Slovakia

Ivan Jarolímek, Jozef Šibík, Mária Šibíková

Institute of Botany, Slovak Academy of Sciences
Dúbravská cesta 9, 845 23 Bratislava, Slovakia

Abstract. In this paper, we present a mathematical model and numerical method designed for the segmentation of satellite images, namely to obtain in an automated way borders of Natura 2000 habitats from Sentinel-2 optical data. The segmentation model is based on the evolving closed plane curve approach in the Lagrangian formulation including the efficient treatment of topological changes. The model contains the term expanding the curve in its outer normal direction up to the region of habitat boundary edges, the term attracting the curve accurately to the edges and the smoothing term given by the influence of local curvature. For the numerical solution, we use the flowing finite volume method discretizing the arising advection-diffusion intrinsic partial differential equation including the asymptotically uniform tangential redistribution of curve grid points. We present segmentation results for satellite data from a selected area of Western Slovakia (Záhorie) where the so-called riparian forests represent the important European Natura 2000 habitat. The automatic segmentation results are compared with the semi-automatic segmentation performed by the botany expert and with the GPS tracks obtained in the field. The comparisons show the ability of our numerical model to segment the habitat areas with the accuracy comparable to the pixel resolution of the Sentinel-2 optical data.

1. **Mathematical model for image segmentation.** In this section, we present the Lagrangian closed plane curve evolution model for the automated segmentation of Natura 2000 habitat areas from Sentinel-2 optical data (satellite images). Natura 2000 is a network of core breeding and resting sites for rare and threatened species. It protects 27,312 sites with terrestrial area 787,606 km^2 (around 18 percent of the land of the EU countries) to ensure the long-term survival of Europe's most valuable species and habitats. Sentinel-2 is an Earth observation mission operated by European Space Agency (ESA) that systematically acquires multiband optical data at high spatial resolution (10 m to 60 m) over land and coastal waters.

2010 *Mathematics Subject Classification.* Primary: 35R01, 65M08; Secondary: 35R37, 92F05.
Key words and phrases. Image segmentation, curve evolution, numerical method, Natura 2000, satellite images, Sentinel-2.

* Corresponding author.

This work was supported by projects APVV-16-0431, APVV-15-0522, VEGA 1/0608/15 and ESA Contract No. 4000122575/17/NL/SC.

In section 2 we show numerical discretization of the suggested model. In section 3 we present the segmentation examples on real satellite images and compare them quantitatively with the semi-automatic segmentation performed by the botany expert and with the GPS tracks obtained in the field. The paper is finished by Conclusions.

We consider the time evolution of a closed planar segmentation curve driven by the following general equation

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{V}(\mathbf{x}, t) \ , \tag{1}$$

where $\mathbf{x}$ denotes a point of the evolving curve in the plane (we call it also the curve position vector), t is the time, $\frac{\partial \mathbf{x}}{\partial t}$ denotes the time derivative (later denoted also by $\mathbf{x}_t$) of the position vector representing the speed of the curve motion, and

$$\mathbf{V}(\mathbf{x}, t) = (1 - \lambda(t))g_2(\mathbf{x})\mathbf{N}(\mathbf{x}, t) - \lambda(t)\nabla g_1(\mathbf{x}) - \delta(t)k(\mathbf{x}, t)\mathbf{N}(\mathbf{x}, t) \tag{2}$$

denotes a suitably designed velocity vector field, ∇ is the gradient operator, $\mathbf{N}$ denotes the outer unit normal vector to the evolving closed curve and k its curvature. The vector field definition contains functions g_1 and g_2, which are used to drive the segmentation curve from its initial position through the segmented region to its border (the term $g_2\mathbf{N}$) and to attract the evolving curve accurately to the border of the segmented region (the term $-\nabla g_1$). Due to a noise and smoothness requirement, the curve evolution is regularized by the curvature term $-k\mathbf{N}$. The time dependent function $\lambda\,(t) \in [0, 1]$ weights the edge attracting vector field $-\nabla g_1$ and the expanding in normal direction velocity $g_2\mathbf{N}$. It may depend on time t, e.g. to diminish expansion in a local edge neighbourhood. Similarly, $\delta(t)$ may weight the curvature influence as time is evolving.

For the edge attracting term of (2) we employ function g_1 defined as follows

$$g_1(\mathbf{x}) = G_{\sigma_1} * g(s, k_1), \quad g(s, k_1) = \frac{1}{1 + k_1 s^2}, \quad s = \left|\nabla G_{\sigma_0} * I^0\right|(\mathbf{x}) \,, \tag{3}$$

where $G_{\sigma_0} * I^0 = I^{\sigma_0}$ represents the Gaussian smoothing (convolution with the Gauss function with variance σ_0) of the original image I^0 rescaled to interval $\langle 0, 1\rangle$ and g is a so-called edge detector function depending on the squared norm of smoothed image intensity gradient and empirical parameter k_1. As seen from the first equation in (3), g can be again smoothed by the Gauss kernel with a variance σ_1 which gives us the final form of edge detector g_1. The edge attracting velocity $-\nabla g_1$ is then constructed at every point $\mathbf{x}$ of the image by using the minus gradient of the edge detector. It drives the evolving curve in a direction towards the edges in the image what can be seen in the form of arrows in Fig. 1, second row right. The importance of this edge attracting vector field was emphasized in [3, 7] where the so-called geodesic active contours were introduced in the level set formulation, and since then it has been used in many successful image segmentation models, e.g. in medicine or biology [18, 14, 4].

The third term in (2), a regularization by curvature with the parameter δ, is used to control the smoothness of the final curve as well as to make the overall curve evolution more regular with respect to noise. However, if we consider only the second and the third terms of (2) and place the initial segmentation curve to the region where the edge attracting vector field is weak, the evolving curve does not move, or even worse, it can shrink to a point because of the influence of curvature. Such a phenomenon must be overcome because, in the case of automatic segmentation,

we used to drive the segmentation curve from inside of the segmented region. To that goal, let us define a function $H(\mathbf{x})$ describing the similarity of image intensity evaluated at $\mathbf{x}$ with the image intensity evaluated in an initial circle. We may consider a vector (list) of features ρ and a vector of allowed (relative or absolute, one-sided or two-sided) deviations from the prescribed feature value which reasonably characterize a segmented region. In the simplest situation, one can consider as a feature just a mean value of the image intensity and then ρ is a scalar parameter. The mean value of image intensity can be defined as the computed average of the image intensity inside the initial segmentation curve or specified externally by a user. Then we may define function H by

$$H(\mathbf{x}) = \begin{cases} 1 & \text{if } I^{\sigma_0}(\mathbf{x}) \in (\rho - \epsilon\rho, \rho + \epsilon\rho) \\ 0 & \text{otherwise}\,. \end{cases} \tag{4}$$

The function H sets value 1 to the pixels inside the regions with intensity similar to ρ and value 0 to pixels in other regions. Another simple option is to compute minimal and maximal image intensity values, ρ_{min}, ρ_{max}, of I^0 or I^{σ_0} inside the initial segmentation curve and define

$$H(\mathbf{x}) = \begin{cases} 1 & \text{if } I^{\sigma_0}(\mathbf{x}) \in (\rho_{min} - \epsilon, \rho_{max} + \epsilon) \\ 0 & \text{otherwise}\,. \end{cases} \tag{5}$$

Of course, it is not necessary to consider just 0 and 1 values in these membership function definitions. It is possible to smooth them by a convolution or consider some fuzzy/probabilistic definitions, but we do not follow such approaches in this paper.

By using definitions (4) or (5), the expanding velocity H would drive the initial curve in its outer normal direction through the segmented region but we have to stop the expansion in a neighbourhood of edges. To that goal we utilize the edge detector g_1 defined in (3) which is close to zero along the edges and construct the final form of the expanding term g_2 by smoothed multiplication of H and g_1 functions,

$$g_2(\mathbf{x}) = G_{\sigma_2} * (H(\mathbf{x})\; g_1(\mathbf{x}))\,. \tag{6}$$

For the illustration of the model inputs (3)-(6) and the model (1)-(2) behaviour, please, look at Fig. 1.

Since in the continuous Lagrangian curve evolution formulation, only the normal component of the velocity determines the shape of the evolving curve (the tangential component just reparametrizes the curve), it is natural to ignore the tangential component of the velocity (2) in the analytical (continuous) formulation of the curve evolution. In the case of (2) there is tangential component only in the edge attracting vector field $-\nabla g_1$ and to remove it we take its projection to the curve normal $\mathbf{N}$ and redefine the basic curve evolution model (1)-(2) to the form

$$\frac{\partial \mathbf{x}}{\partial t} = \beta \mathbf{N}\;, \tag{7}$$

where the so-called outer normal velocity β is defined by

$$\beta = (1 - \lambda\,(t))g_2 - \lambda\,(t)\,\nabla g_1 \cdot \mathbf{N} - \delta k\,. \tag{8}$$

On the other hand, from the numerical point of view it is necessary to add suitable velocity of grid points in tangential direction $\mathbf{T}$ in order to prevent numerically evolving curve from selfintersections and other singularities and also to allow fast detection and solution of topological changes (splitting and merging of curves), see

[6, 8, 11, 12, 13, 14, 1]. So we enrich equation (7) by the asymptotically uniform tangential redistribution of points [12] and consider (7) in the new form

$$\frac{\partial \mathbf{x}}{\partial t} = \beta \mathbf{N} + \alpha \mathbf{T} \tag{9}$$

with the normal velocity β given by (8) and with the tangential velocity α defined in section 2.1.

2. Numerical algorithm for evolving curves.

Let Γ be a closed plane curve, $\Gamma : S^1 \to \mathbf{R}^2$, parametrized by $u \in S^1$, where S^1 is a circle with unit length, thus $u \in [0,1]$ and $\Gamma = \{\mathbf{x}(u), u \in S^1\}$, where $\mathbf{x}(u) = (\mathbf{x}_1(u), \mathbf{x}_2(u))$ is position vector of the curve Γ for parameter u. In the sequel, the curve will be discretized to a set of points. An example of a closed planar curve discretization is displayed in Fig. 2, where $\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_n$ are discrete curve points which correspond to the uniform discretization of the unit circle with step $h = 1/n$ and $\mathbf{x}_0 = \mathbf{x}_n$.

Let $|\mathbf{x}_u| > 0$, where $\mathbf{x}_u = \left(\frac{\partial \mathbf{x}_1}{\partial u}, \frac{\partial \mathbf{x}_2}{\partial u}\right)$ and $g = |\mathbf{x}_u| = \sqrt{\left(\frac{\partial \mathbf{x}_1}{\partial u}\right)^2 + \left(\frac{\partial \mathbf{x}_2}{\partial u}\right)^2}$. Let us denote by s the unit arc-length parametrization of the curve Γ. Then $ds = |\mathbf{x}_u| du = g du$ and $du = \frac{1}{g} ds$. If the curve Γ is parametrized in a counter-clockwise direction, the unique definition of the unit tangent $\mathbf{T}$ and (outer) normal $\mathbf{N}$ vectors to the curve Γ can be done as follows: $\mathbf{T} = \frac{\partial \mathbf{x}}{\partial s}$ (denoted also by $\mathbf{x}_s$), $\mathbf{N} = \mathbf{x}_s^{\perp}$ and $\mathbf{T} \wedge \mathbf{N} = -1$, where $\mathbf{T} \wedge \mathbf{N}$ denotes the determinant of the matrix with columns $\mathbf{T}$ and $\mathbf{N}$. If $\mathbf{T} = (\mathbf{x}_{1s}, \mathbf{x}_{2s})$, then $\mathbf{N} = (\mathbf{x}_{2s}, -\mathbf{x}_{1s})$. Since we consider outer normal, from the Frenet-Serret formulas we have $\mathbf{T}_s = -k\mathbf{N}$ and $\mathbf{N}_s = k\mathbf{T}$, where k is the curvature. From there it follows that $-k\mathbf{N} = \mathbf{T}_s = (\mathbf{x}_s)_s = \mathbf{x}_{ss}$.

In our approach, the curve Γ is given by its position vector $\mathbf{x}$, so its evolution can be described by the evolution of this vector in time. We consider the general form of the curve evolution (9) where $\beta = w - \delta k$ with

$$w = (1 - \lambda(t)) g_2 - \lambda(t) \nabla g_1 \cdot \mathbf{N}. \tag{10}$$

Then using the Frenet-Serret formula, mentioned above, we can rewrite (9) into the form of so-called intrinsic partial differential equation

$$\mathbf{x}_t = (w - \delta k)\mathbf{N} + \alpha \mathbf{T} = w\mathbf{N} - \delta k \mathbf{N} + \alpha \mathbf{T} = \delta \mathbf{x}_{ss} + \alpha \mathbf{x}_s + w \mathbf{x}_s^{\perp} \tag{11}$$

which is suitable for numerical discretization. Since $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$, (11) represents a system of two partial differential equations for components $\mathbf{x}_1$ and $\mathbf{x}_2$ of the curve position vector $\mathbf{x}$. These two equations are coupled together by the derivatives with respect to the arc-length parametrization s, because both components of the position vector $\mathbf{x}$ occur in the term ds. The curvature term yields the so-called intrinsic diffusion along the curve (the term $\mathbf{x}_{ss}$), the tangential velocity yields the so-called intrinsic advection along the curve (the term $\mathbf{x}_s$) and the external driving force in the normal direction is given by the third term on the right hand side of (11).

2.1. Suitable choice of tangential velocity.

Although it is well-known that a tangential motion does not change the shape of the evolving curve, we know that it is helpful in a stabilization of the numerical algorithms based on Lagrangian approaches [11, 12]. If we want to redistribute the points along the curve, we have to study the ratio $\frac{g}{L}$, where $g = |\mathbf{x}_u|$ represents the local and L the global curve

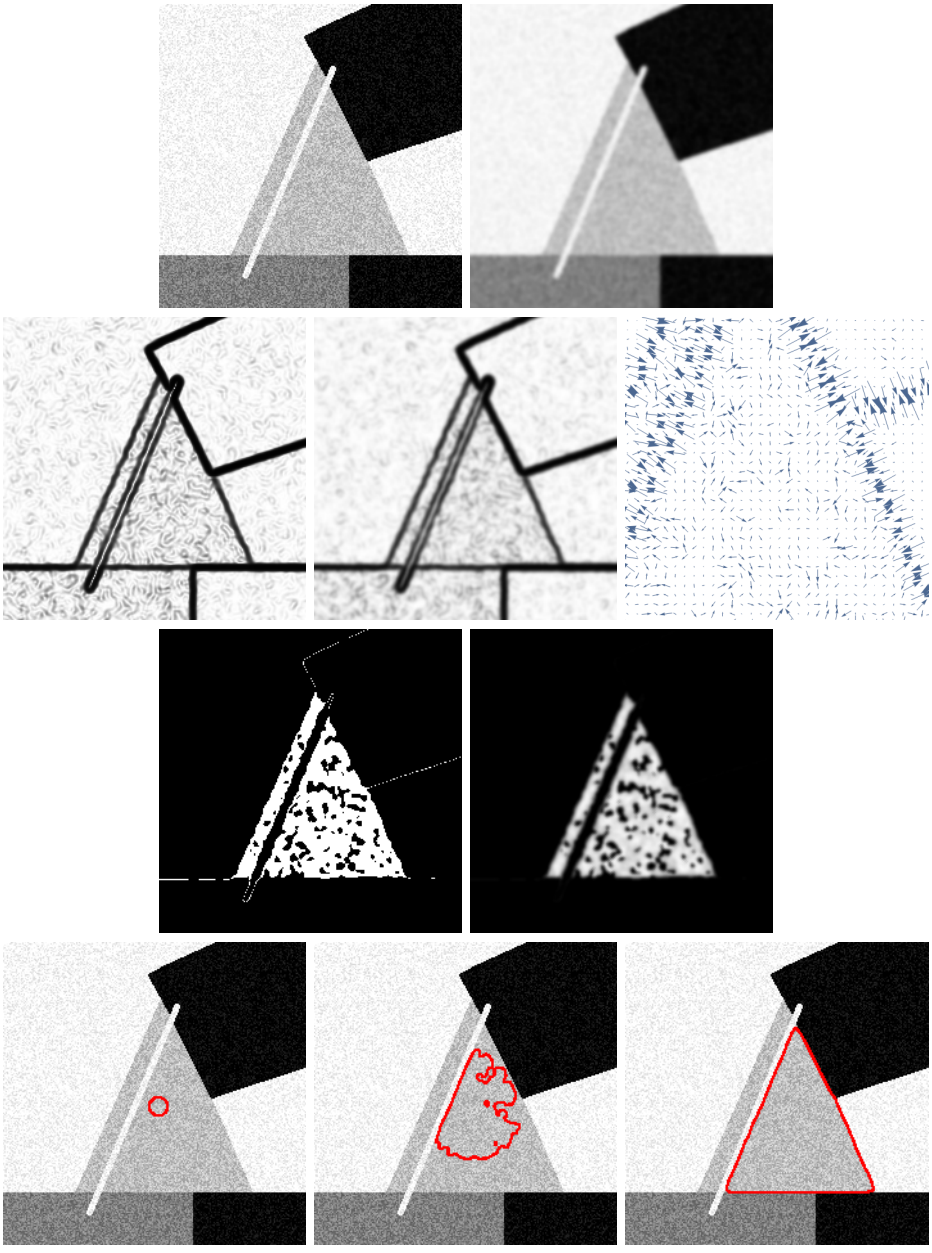

FIGURE 1. First row: the original image I^0 and smoothed image I^{σ_0}. Second row: the visualization of $g(\mathbf{x})$, smoothed edge detector $g_1(\mathbf{x})$ and a zoom of the vector field $-\nabla g_1(\mathbf{x})$ where we see arrows pointing towards the edges in I^0. Third row: the functions $H(\mathbf{x})$ using (5) and $g_2(\mathbf{x})$ evaluated by using the initial circle plotted in the Fourth row, left. Fourth row: the initial segmentation curve placed in I^0 and its time evolution until the final segmentation state (bottom right). In the middle image we see that the evolving curve undergoes topological changes which are resolved efficiently.

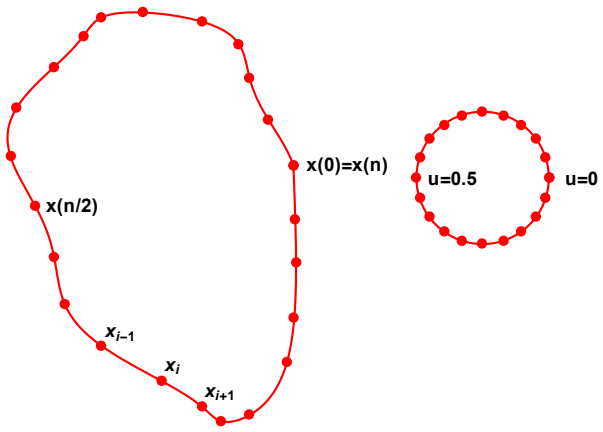

FIGURE 2. Closed planar curve discretization (left) corresponding to the uniform discretization of the unit circle (right).

lengths. Considering this ratio in a discrete form, we obtain

$$\frac{g}{L} \approx \frac{\frac{|\mathbf{x}_i - \mathbf{x}_{i-1}|}{h}}{L} = \frac{|\mathbf{x}_i - \mathbf{x}_{i-1}|}{Lh} = \frac{|\mathbf{x}_i - \mathbf{x}_{i-1}|}{\frac{L}{n}}, \tag{12}$$

where $h = \frac{1}{n}$, n is the number of the curve grid points, see Figure 2. The numerator represents a real distance between two neighbouring points, and the denominator denotes the uniform distance between neighbouring points. It is clear that one can get the curve with uniformly distributed discrete grid points, if this ratio tends to 1, i.e. $\frac{|\mathbf{x}_i - \mathbf{x}_{i-1}|}{\frac{L}{n}} \to 1$ for all discrete curve segments. Thus, in continuous formulation we need to fulfill the condition $\frac{g}{L} \to 1$.

For the time evolution of the curve local length g, by using the Frenet-Serret formulas, we obtain subsequently

$$\begin{aligned} g_t = |\mathbf{x}_u|_t &= \frac{\mathbf{x}_u}{|\mathbf{x}_u|} \cdot (\mathbf{x}_u)_t = \frac{g\mathbf{x}_s}{g} \cdot (\mathbf{x}_t)_u = \mathbf{T} \cdot (\beta\mathbf{N} + \alpha\mathbf{T})_u = \mathbf{T} \cdot g\,(\beta\mathbf{N} + \alpha\mathbf{T})_s \\ &= \mathbf{T} \cdot g\,(\beta_s\mathbf{N} + \beta\mathbf{N}_s + \alpha_s\mathbf{T} + \alpha\mathbf{T}_s) = \mathbf{T} \cdot g\,(\beta_s\mathbf{N} + \beta k\mathbf{T} + \alpha_s\mathbf{T} - \alpha k\mathbf{N}) \\ &= \mathbf{T} \cdot g\,((\beta_s - \alpha k)\,\mathbf{N} + (\beta k + \alpha_s)\,\mathbf{T}) = gk\beta + g\alpha_s = gk\beta + \alpha_u. \end{aligned} \tag{13}$$

By integrating the previous equation, we obtain the formula for the evolution of the global curve length L

$$L_t = \int_0^1 g_t du = \int_0^1 gk\beta du + \int_0^1 \alpha_u du = \int_0^1 gk\beta du + \alpha\,(1) - \alpha\,(0)\,. \tag{14}$$

Due to the periodic boundary condition we have $\alpha\,(1) = \alpha\,(0)$ and

$$L_t = \int_0^1 gk\beta du = \int_\Gamma k\beta\, ds = L\,\langle k\beta\rangle_\Gamma\,, \tag{15}$$

where $\langle k\beta\rangle_\Gamma = \frac{1}{L}\int_\Gamma k\beta\, ds$. For the time evolution of the ratio $\frac{g}{L}$ we then obtain

$$\left(\frac{g}{L}\right)_t = \frac{(gk\beta + \alpha_u)\, L - gL\,\langle k\beta\rangle_\Gamma}{L^2} = \frac{g}{L}\left(k\beta + \alpha_s - \langle k\beta\rangle_\Gamma\right). \tag{16}$$

If we now impose that $\left(\frac{g}{L}\right)_t = \omega\left(1 - \frac{g}{L}\right)$, where ω is a parameter determining how fast the redistribution becomes uniform, we get desired condition $\frac{g}{L} \to 1$, and we obtain the formula for the tangential velocity which guarantees the asymptotically uniform redistribution of grid points

$$\alpha_s = \langle k\beta\rangle_\Gamma - k\beta + \omega\left(\frac{L}{g} - 1\right). \tag{17}$$

2.2. **Numerical discretization.** Let us recall the intrinsic PDE (11) for the planar curve evolution and write it in the following form

$$\mathbf{x}_t - \alpha\mathbf{x}_s = \delta\mathbf{x}_{ss} + w\mathbf{x}_s^\perp, \tag{18}$$

where w and α are given by (10) and (17). First, we perform the spatial discretization, which is based on the flowing finite volume method [11, 13] and then we discuss the time discretization, which is semi-implicit in the intrinsic diffusion term and uses inflow-implicit/outflow-explicit strategy for intrinsic advection term [11, 13, 9, 10].

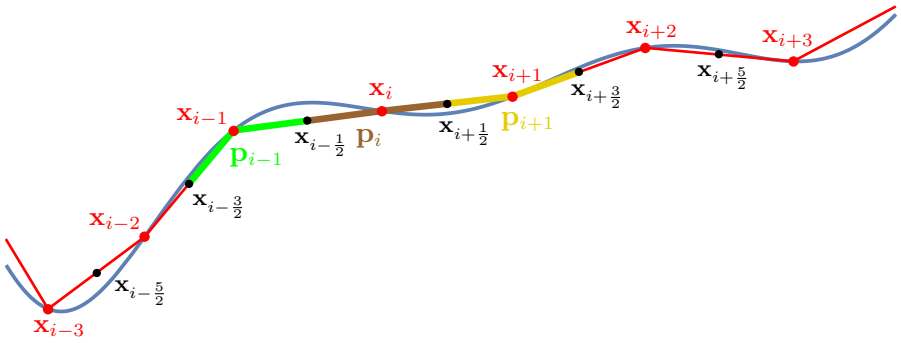

FIGURE 3. Visualization of the curve discretization: curve grid points (red) and their midpoints. Finite volumes $\mathbf{p}_{i-1}, \mathbf{p}_i$, and $\mathbf{p}_{i+1}$ are highlighted by green, brown and yellow color. Note that $\mathbf{p}_i$ is not a straight line given by $\mathbf{x}_{i-\frac{1}{2}}$ and $\mathbf{x}_{i+\frac{1}{2}}$, but a broken line given by $\mathbf{x}_{i-\frac{1}{2}}$, $\mathbf{x}_i$ and $\mathbf{x}_{i+\frac{1}{2}}$, see also [1].

Integrating (18) over the finite volume $\mathbf{p}_i = [\mathbf{x}_{i-\frac{1}{2}}, \mathbf{x}_{i+\frac{1}{2}}]$, see Fig. 3, where $\mathbf{x}_{i-\frac{1}{2}}$ represents the middle point between the points $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$, i.e.

$$\mathbf{x}_{i-\frac{1}{2}} = \frac{\mathbf{x}_{i-1} + \mathbf{x}_i}{2}, \tag{19}$$

we get

$$\int_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \mathbf{x}_t ds - \alpha \int_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \mathbf{x}_s ds = \delta \int_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \mathbf{x}_{ss} ds + w \int_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \mathbf{x}_s^\perp ds, \tag{20}$$

where the values δ, α and w are considered constant, with values δ_i, α_i, w_i on the discrete curve segment $\mathbf{p}_i$ around the point $\mathbf{x}_i$. We define $h_i = |\mathbf{x}_i - \mathbf{x}_{i-1}|$, then the measure of the segment $\mathbf{p}_i$ is equal to $\frac{h_i+h_{i+1}}{2}$, and using the Newton-Leibniz formula we get the following approximation of (20), for $i = 1, ..., n$,

$$\frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t - \alpha_i [\mathbf{x}]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} = \delta_i [\mathbf{x}_s]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} + w_i \left([\mathbf{x}]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \right)^{\perp} \tag{21}$$

$$\frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t - \alpha_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right) = \delta_i [\mathbf{x}_s]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} + w_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right)^{\perp}. \tag{22}$$

Approximating the arc-length derivative $\mathbf{x}_s$ in the first bracket on the right hand side by a finite difference we obtain the semi-discrete flowing finite volume scheme,

$$\begin{aligned} \tfrac{h_i+h_{i+1}}{2} (\mathbf{x}_i)_t - \alpha_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right) &= \delta_i \left(\tfrac{\mathbf{x}_{i+1}-\mathbf{x}_i}{h_{i+1}} - \tfrac{\mathbf{x}_i-\mathbf{x}_{i-1}}{h_i} \right) \\ &+ w_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right)^{\perp}. \end{aligned} \tag{23}$$

Now we split the advection term (involving the tangential velocity α) as follows,

$$-\alpha_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right) = \alpha_i \left(\mathbf{x}_i - \mathbf{x}_{i+\frac{1}{2}} \right) - \alpha_i \left(\mathbf{x}_i - \mathbf{x}_{i-\frac{1}{2}} \right). \tag{24}$$

Using (19) we obtain

$$\begin{aligned} &\frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t + \frac{\alpha_i}{2} (\mathbf{x}_i - \mathbf{x}_{i+1}) - \frac{\alpha_i}{2} (\mathbf{x}_i - \mathbf{x}_{i-1}) \\ &= \delta_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h_{i+1}} - \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{h_i} \right) + w_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_{i-1}}{2} \right)^{\perp}. \end{aligned} \tag{25}$$

If $\alpha_i < 0$, i.e. the velocity of advection $(-\alpha)$ in (18) is positive in the segment p_i, there is an inflow into the finite volume through its boundary point $\mathbf{x}_{i-\frac{1}{2}}$ and an outflow through the boundary point $\mathbf{x}_{i+\frac{1}{2}}$. On the other hand, if $\alpha_i > 0$, there is an inflow in $\mathbf{x}_{i+\frac{1}{2}}$ and an outflow in $\mathbf{x}_{i-\frac{1}{2}}$. Let us define

$$b_{i-\frac{1}{2}}^{in} = max(-\alpha_i, 0), b_{i-\frac{1}{2}}^{out} = min(-\alpha_i, 0),$$

$$b_{i+\frac{1}{2}}^{in} = max(\alpha_i, 0), b_{i+\frac{1}{2}}^{out} = min(\alpha_i, 0) \tag{26}$$

and rewrite the equation (25) as follows

$$\begin{aligned} &\frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t + \frac{1}{2} \left(b_{i+\frac{1}{2}}^{in} + b_{i+\frac{1}{2}}^{out} \right) (\mathbf{x}_i - \mathbf{x}_{i+1}) + \frac{1}{2} \left(b_{i-\frac{1}{2}}^{in} + b_{i-\frac{1}{2}}^{out} \right) (\mathbf{x}_i - \mathbf{x}_{i-1}) \\ &= \delta_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h_{i+1}} - \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{h_i} \right) + w_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_{i-1}}{2} \right)^{\perp}. \end{aligned} \tag{27}$$

In order to perform the time discretization, let us denote by m the time step index and by τ the length of the discrete time step. Let us approximate the time derivative by the finite difference $(\mathbf{x}_i)_t = \frac{\mathbf{x}_i^{m+1} - \mathbf{x}_i^m}{\tau}$. Let us take the unknowns in the inflow part of the advection term implicitly and in the outflow part explicitly. Then approximate also the expanding term explicitly and the diffusion term implicitly. In that (semi-implicit) way we obtain the fully discrete scheme in the form of a

cyclic tridiagonal system

$$\begin{aligned}
&\mathbf{x}_{i-1}^{m+1}\left(-\frac{\delta_i^m}{h_i^m}-\frac{{b^{in}}^m_{i-\frac{1}{2}}}{2}\right)+\mathbf{x}_{i+1}^{m+1}\left(-\frac{\delta_i^m}{h_{i+1}^m}-\frac{{b^{in}}^m_{i+\frac{1}{2}}}{2}\right)\\
+\mathbf{x}_i^{m+1}&\left(\frac{h_{i+1}^m+h_i^m}{2\tau}+\frac{\delta_i^m}{h_i^m}+\frac{\delta_i^m}{h_{i+1}^m}+\frac{{b^{in}}^m_{i-\frac{1}{2}}}{2}+\frac{{b^{in}}^m_{i+\frac{1}{2}}}{2}\right)=\mathbf{x}_i^m\frac{h_{i+1}^m+h_i^m}{2\tau}\\
&-\frac{{b^{out}}^m_{i+\frac{1}{2}}}{2}\left(\mathbf{x}_i^m-\mathbf{x}_{i+1}^m\right)-\frac{{b^{out}}^m_{i-\frac{1}{2}}}{2}\left(\mathbf{x}_i^m-\mathbf{x}_{i-1}^m\right)+w_i^m\left(\frac{\mathbf{x}_{i+1}^m-\mathbf{x}_{i-1}^m}{2}\right)^{\perp},
\end{aligned}\tag{28}$$

$i=1,...,n$, where n is the number of the curve grid points.

In the segmentation process the evolving curve can split due to a noisy data or merge with another segmentation curve and thus it can have locally sharp corners or other singular points. In such singular points we modify the scheme (28) and use just the first order implicit upwind scheme instead of the second order inflow-implicit/outflow-explicit method in the discretization of the advection term. This first order upwind scheme is applied when the angle between two consecutive curve segments $(\mathbf{x}_{i-1},\mathbf{x}_i)$ and $(\mathbf{x}_i,\mathbf{x}_{i-1})$ is less than 120°, i.e. $\measuredangle(\mathbf{x}_{i-1}\mathbf{x}_i\mathbf{x}_{i+1})<120°$. In the upwind scheme we use only the inflow velocities $b^{in}_{i-\frac{1}{2}}$, $b^{in}_{i+\frac{1}{2}}$, and points $\mathbf{x}_{i-\frac{1}{2}}$, $\mathbf{x}_{i+\frac{1}{2}}$ in (24) are approximated by the neighbouring values $\mathbf{x}_{i-1}$ or $\mathbf{x}_{i+1}$, depending on the inflow direction. In this way we get

$$\begin{aligned}
&\frac{h_i+h_{i+1}}{2}\left(\mathbf{x}_i\right)_t+b^{in}_{i+\frac{1}{2}}\left(\mathbf{x}_i-\mathbf{x}_{i+1}\right)+b^{in}_{i-\frac{1}{2}}\left(\mathbf{x}_i-\mathbf{x}_{i-1}\right)\\
&=\delta_i\left(\frac{\mathbf{x}_{i+1}-\mathbf{x}_i}{h_{i+1}}-\frac{\mathbf{x}_i-\mathbf{x}_{i-1}}{h_i}\right)+w_i\left(\frac{\mathbf{x}_{i+1}-\mathbf{x}_{i-1}}{2}\right)^{\perp}
\end{aligned}\tag{29}$$

instead of (27) and in a singular point we use

$$\begin{aligned}
&\mathbf{x}_{i-1}^{m+1}\left(-\frac{\delta_i^m}{h_i^m}-{b^{in}}^m_{i-\frac{1}{2}}\right)+\mathbf{x}_{i+1}^{m+1}\left(-\frac{\delta_i^m}{h_{i+1}^m}-{b^{in}}^m_{i+\frac{1}{2}}\right)\\
+\mathbf{x}_i^{m+1}&\left(\frac{h_{i+1}^m+h_i^m}{2\tau}+\frac{\delta_i^m}{h_i^m}+\frac{\delta_i^m}{h_{i+1}^m}+{b^{in}}^m_{i-\frac{1}{2}}+{b^{in}}^m_{i+\frac{1}{2}}\right)=\mathbf{x}_i^m\frac{h_{i+1}^m+h_i^m}{2\tau}\\
&+w_i^m\left(\frac{\mathbf{x}_{i+1}^m-\mathbf{x}_{i-1}^m}{2}\right)^{\perp}
\end{aligned}\tag{30}$$

instead of (28). This replacement of (28) by (30) occurs rarely, but in case it arises, the usage of (30) makes the scheme robust with respect to singularities.

The system (28), (30) is always represented by a strictly diagonally dominant matrix, thus it is always solvable by the efficient cyclic tridiagonal solver (a modification of the basic Thomas algorithm) and the solvability is guaranteed without any restriction on time step length τ [14].

In the numerical scheme (28),(30) there are two parameters α_i^m and w_i^m, given by (17) and (10), which are evaluated as follows,

$$w_i^m=\left(\left(1-\lambda_i^m\right)g_{2i}-\lambda_i^m(\nabla g_1)_i\cdot\mathbf{N}_i^m\right),\tag{31}$$

where $\mathbf{N}_i^m=\left(\frac{\mathbf{x}_{i+1}^m-\mathbf{x}_{i-1}^m}{h_{i+1}^m+h_i^m}\right)^{\perp}$ and g_{2i} and $(\nabla g_1)_i$ are discrete values of the functions g_2 and ∇g_1 in the curve point $\mathbf{x}_i^m$. The partial derivatives of g_1 included in the gradient are evaluated by finite differences on the pixel grid.

In order to discretize the tangential velocity α we first set $\alpha_0^m = 0$, which causes the point $\mathbf{x}_0$ will move only in the normal direction. Then, by the finite volume discretization of equation (17) we get α_i^m for $i = 1, 2, ..., n-1$ by

$$\alpha_i^m = \alpha_{i-1}^m + h_i^m \langle k\beta \rangle_\Gamma^m - h_i^m k_i^m \beta_i^m + \omega \left(\frac{L^m}{n} - h_i^m \right), \tag{32}$$

where the curvature k_i^m, the normal velocity β_i^m, for $i = 1, 2, ..., n$, the mean value $\langle k\beta \rangle_\Gamma$ and the total curve length L^m are given by following formulas

$$k_i^m = \operatorname{sgn}\left(\mathbf{h}_{i-1}^m \wedge \mathbf{h}_{i+1}^m\right) \frac{1}{2h_i^m} \arccos\left(\frac{\mathbf{h}_{i+1}^m \cdot \mathbf{h}_{i-1}^m}{h_{i+1}^m h_{i-1}^m} \right), \tag{33}$$

$$\beta_i^m = -\frac{\delta_{i-1}^m + \delta_i^m}{2} k_i^m + \frac{w_{i-1}^m + w_i^m}{2}, \tag{34}$$

$$\langle k\beta \rangle_\Gamma = \frac{1}{L^m} \sum_{l=1}^{n} h_l^m k_l^m \beta_l^m, \tag{35}$$

$$L^m = \sum_{l=1}^{n} h_l^m, \tag{36}$$

where $\mathbf{h}_i^m = \mathbf{x}_i^m - \mathbf{x}_{i-1}^m$, $h_i^m = |\mathbf{h}_i^m|$.

Since the initial setting for $\alpha_0^m = 0$ can cause unnecessary large values of the tangential velocity in an effort to redistribute the curve points uniformly, we find the average tangential velocity $\alpha_{avg}^m = \sum_{i=0}^{n} \frac{\alpha_i^m}{n}$. It is clear that α_{avg}^m is an unnecessary tangential velocity and therefore we find new minimized values as $\overline{\alpha_i^m} = \alpha_i^m - \alpha_{avg}^m$ for $i = 0, 1, ..., n-1$, by which we redefine α_i^m and use these new values inside the scheme (28),(30).

The last issue that must be solved in the numerical implementation of the Lagrangian curve evolution approach is topological changes treatment. By a topological change, we mean merging of several evolving curves and/or splitting of the evolving curve into several separate curves. The splitting can occur when the curve velocity is locally slowed down significantly and merging when, e.g., we begin with several initial curves. Detecting and solving the topological changes in the Lagrangian approach is usually highly time-consuming because the standard approaches have computational complexity $O\left(n^2\right)$, where n is the number of curve points [5, 17]. Such a high complexity is due to the strategy for the topological changes detection, which consists of computing pairwise distances between all grid points of the curve. The number of operations in such an approach is proportional to $(n-1)^2 = n^2 - 2n + 1$ and such a number of additional operations should be performed in every computational time step to evolve the curve. In general, it would slow down the overall computational time significantly and so one has to implement a different, efficient solution. We took our solution from [2, 14, 1] where the topological changes are detected and resolved with $O(n)$ complexity, which gives real-time fast algorithm and makes our Lagrangian approach for image segmentation efficient and reliable.

3. **Numerical experiments.**

In the following experiments, we applied the proposed segmentation method to the real Sentinel-2 optical data and we have been testing the method on selected Natura 2000 habitat - *91F0 Riparian mixed forests along the great rivers* - in Záhorie region of Western Slovakia. We used Sentinel-2

data from April 21, 2018 and its B03-Green (Figs. 4-7) and B04-Red (Fig. 8) optical bands (with 2.5% cropped histogram) as the image intensity function I^0. The reason to use these channels is that in this period of the year the riparian forests are best observable/distinguishable in these optical bands. Before applying the functions g_1, H and g_2 we always rescale the cropped image intensity I^0 to the interval $[0,1]$.

The first two numerical experiments serve for illustration of the method behaviour on real data. The function H was given by (5) with small $\epsilon = 0.025$ in manually placed initial circle. Further parameters, σ_l, $l = 0,1,2$ in convolutions and k_1 in definition of g_1, were chosen according to the smoothness of the original data in expected segmented regions and along their boundaries. The weight λ was set to 0.5 until the overall normal motion of the curve was slowed down to $\beta < 0.001$ and then it was set to 1. The curvature influence parameter δ was considered small, $\delta = 0.01$. For the obtained segmentation results and the visualization of the method inputs, look at Figs. 4-5. Note, that $g_2(\mathbf{x})$ is the term expanding the curve. Curvature term is helpful in case of inhomogeneity of $g_2(\mathbf{x})$ in the segmented area. It allows the curve to expand to the boundary also through inhomogeneities in $g_2(\mathbf{x})$, see Figs. 4 and 5, third row right. Near the boundary, the expanding term weakens and the edge detector attracts the curve to the edge. One can see in the second numerical experiment that we can begin with two separate initial curves which are subsequently merged and found the final riparian forest segmentation. From the CPU time point of view, the method is fast, the computations took 0.45 and 2.6 seconds for these two numerical experiments.

In the next Figs. 6-8 we are comparing the results of the automatic segmentation of riparian forests in Záhorie region with the GPS tracks obtained in the field and with the semi-automatic segmentations obtained by the botany expert using the method [15]. In Figs. 6-8 we placed the evolving curves into the classical RGB images to see better the desired segmented regions. The segmentation parameters were chosen similarly to experiments in Figs. 4-5, but the function H was given by (5) for Fig. 7 and by (4) for the other two figures, δ was chosen 0.1 until the curve was slowed down and then δ was set to 1 in all three experiments.

For the comparison we are using the mean and maximal Hausdorff distances. The mean Hausdorff distance $\overline{\mathrm{d}}_H(A,B)$ of two discrete curves is given by the following formula

$$\overline{\delta}_H(A,B) = \frac{1}{n}\sum_{i=1}^{n} \min_{b\in B} d(a_i,b), \overline{\delta}_H(B,A) = \frac{1}{m}\sum_{i=1}^{m} \min_{a\in A} d(a,b_i), \tag{37}$$

$$\overline{\mathrm{d}}_H(A,B) = \frac{\overline{\delta}_H(A,B) + \overline{\delta}_H(B,A)}{2}, \tag{38}$$

where $d(a_i,b)$ is the Euclidean distance of two points a_i and b from the point sets $A = \{a_1,a_2,a_3,...,a_n\}$ and $B = \{b_1,b_2,b_3,...,b_m\}$. The maximal (or classical) Hausdorff distance $\mathrm{d}_H(A,B)$ is given by

$$\mathrm{d}_H(A,B) = \max\left\{\sup_{a\in A}\inf_{b\in B} d(a,b), \sup_{b\in B}\inf_{a\in A} d(a,b)\right\}. \tag{39}$$

The quantitative comparisons are as follows:

Fig. 6: The mean Hausdorff distance of the automatic and semi-automatic segmentation is 6.83m and their maximal Hausdorff distance is 27.20m. The mean

Hausdorff distance of the automatic segmentation and GPS track obtained by botanists in the field is 12.42m and their maximal Hausdorff distance is 39.21m.

Fig. 7: The mean Hausdorff distance of the automatic and semi-automatic segmentation is 10.73m and their maximal Hausdorff distance is 63.95m. The mean Hausdorff distance of the automatic segmentation and GPS track obtained by botanists in the field is 10.82m and their maximal Hausdorff distance is 50.44m.

Fig. 8: The mean Hausdorff distance of the automatic and semi-automatic segmentation is 8.48m and their maximal Hausdorff distance is 51.85m. The mean Hausdorff distance of the automatic segmentation and GPS track obtained by botanists in the field is 11.70m and their maximal Hausdorff distance is 49.96m.

As one can see, the mean Hausdorff distances of all compared curves are comparable to Sentinel-2 pixel resolution accuracy (10m) and the maximal ones are in the range of few tens of meters, which indicate desired accuracy of developed automatic segmentation method.

4. **Conclusions.**

In this paper, we presented the design and numerical implementation of the method for the automatic segmentation of Natura 2000 habitats from Sentinel-2 optical data. We discussed details of mathematical model design and its numerical discretization including efficient and robust computer implementation. The CPU times of the method are in the range of few seconds (or less) and presented tests show the accuracy (in the sense of the mean Hausdorff distance) in the range of Sentinel-2 pixel resolution (10m). The method was tested on selected Natura 2000 habitat, the riparian forests, where we have at disposal semi-automatic segmentation performed by botany experts as well as the GPS tracks obtained by botanists in the field.

Future usage of the developed automatic segmentation method will be twofold, first for finding occurrences and borders of habitats and second for monitoring the habitat changes in time. The second goal is very important from the society point of view. We will start with already performed accurate habitat area segmentation and evolve automatically such initially given segmentation curve in a newly incoming satellite data to check whether it undergoes significant change. Such a change would indicate the change in habitat area and/or conditions and should lead to actions to protect the Nature reserve. Since the segmentation method is computationally fast, we will be able to use the inverse modelling [16] to find optimal segmentation parameters for every habitat area where we have at disposal GPS tracks and/or semi-automatic segmentation result. The optimal parameters will be chosen with the goal to minimize the mean Hausdorff distance of the automatic segmentation and GPS track/semi-automatic segmentation.

REFERENCES

[1] M. Ambroz, M. Balažovjech, M. Medla and K. Mikula, Numerical modeling of wildland surface fire propagation by evolving surface curves, *Adv. Comput. Math.*, **45** (2019), 1067–1103.

[2] M. Balažovjech, K. Mikula, M. Petrášová and J. Urbán, Lagrangean method with topological changes for numerical modelling of forest fire propagation, 19th Conference on Scientific Computing, Slovakia, 2012.

[3] V. Caselles, R. Kimmel and G. Sapiro, Geodesic active contours, *Internat. J. Comput. Vision*, **22** (1997), 61–79.

[4] E. Faure et al., A workflow to process 3D+time microscopy images of developing organisms and reconstruct their cell lineage, *Nat. Commun.*, **7** (2016).

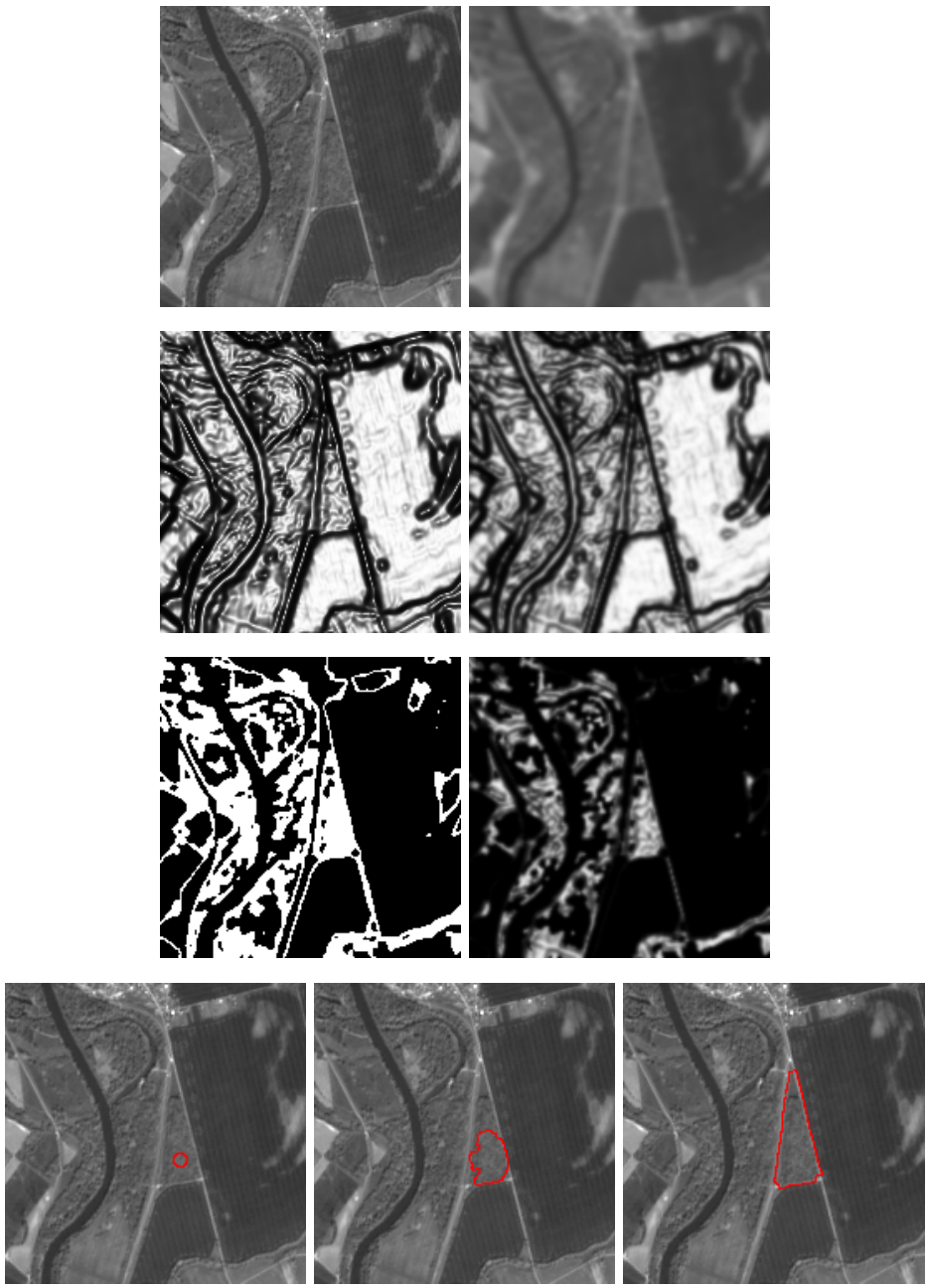

FIGURE 4. First row: the original image I^0 and smoothed image I^{σ_0}. Second row: the visualization of the function $g(\mathbf{x})$ and smoothed edge detector $g_1(\mathbf{x})$. Third row: the function $H(\mathbf{x})$ and $g_2(\mathbf{x})$ evaluated by using the initial circle plotted in the Fourth row, left. Fourth row: the initial segmentation curve placed in I^0 (bottom left) and its time evolution (bottom middle) until the final segmentation state (bottom right) is reached.

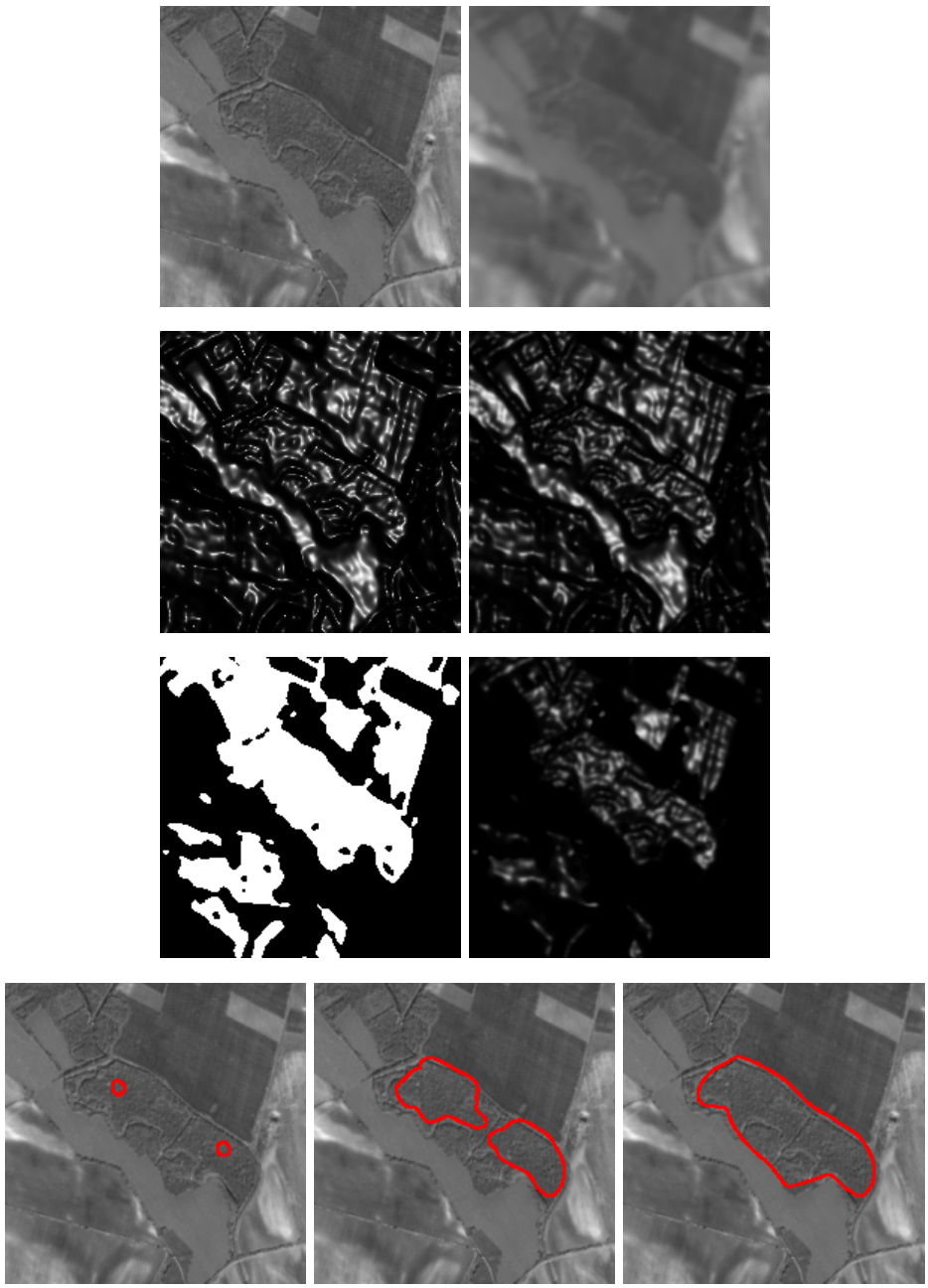

FIGURE 5. First row: the original image I^0 and smoothed image I^{σ_0}. Second row: the visualization of the function $g(\mathbf{x})$ and smoothed edge detector $g_1(\mathbf{x})$. Third row: the function $H(\mathbf{x})$ and $g_2(\mathbf{x})$ evaluated by using the initial circle plotted in the Fourth row, left. Fourth row: the initial segmentation curve placed in I^0 (bottom left) and its time evolution (bottom middle) until the final segmentation state (bottom right) is reached.

FIGURE 6. Left: the evolution of the segmentation curve from the initial circle to the final state. Right: the final automatic segmentation (red) together with the result of the semi-automatic segmentation (yellow) and the GPS track (light-blue).

FIGURE 7. Left: the evolution of the segmentation curve from the initial circle to the final state. Right: the final automatic segmentation (red) together with the result of the semi-automatic segmentation (yellow) and the GPS track (light-blue).

FIGURE 8. Left: the evolution of the segmentation curve from the initial circle to the final state. Right: the final automatic segmentation (red) together with the result of the semi-automatic segmentation (yellow) and the GPS track (light-blue).

[5] M.A. Finney, et al., FARSITE: Fire Area Simulator–model development and evaluation, U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station, Ogden, UT, 1998.

[6] T.Y. Hou, J. Lowengrub and M. Shelley, Removing the stiffness from interfacial flows with surface tension, *J. Comput. Phys.*, **114** (1994), 312–338.

[7] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum and A. Yezzi, Conformal curvature flows: From phase transitions to active vision, *Arch. Rational Mech. Anal.*, **134** (1996), 275–301.
[8] M. Kimura, Numerical analysis for moving boundary problems using the boundary tracking method, *Japan J. Indust. Appl. Math.*, **14** (1997), 373–398.
[9] K. Mikula and M. Ohlberger, Inflow-implicit/outflow-explicit scheme for solving advection equations, in *Finite Volumes for Complex Applications VI. Problems & Perspectives. Volume 1, 2*, Springer Proc. Math., 4, Springer, Heidelberg, 2011, 683–691.
[10] K. Mikula, M. Ohlberger and J. Urbán, Inflow-implicit/outflow-explicit finite volume methods for solving advection equations, *Appl. Numer. Math.*, **85** (2014), 16–37.
[11] K. Mikula and D. Ševčovič, Evolution of plane curves driven by a nonlinear function of curvature and anisotropy, *SIAM J. Appl. Math.*, **61** (2001), 1473–1501.
[12] K. Mikula and D. Ševčovič, A direct method for solving an anisotropic mean curvature flow of plane curves with an external force, *Math. Methods Appl. Sci.*, **27** (2004), 1545–1565.
[13] K. Mikula, D. Ševčovič, M.Balažovjech, A simple, fast and stabilized flowing finite volume method for solving general curve evolution equations, *Commun. Comput. Phys.*, **7** (2010), 195–211.
[14] K. Mikula and J. Urbán, New fast and stable Lagrangean method for image segmentation, 5th International Congress on Image and Signal Processing, Chongqing, China, 2012.
[15] K. Mikula, et al., Report on semi-automatic segmentation methods and software tool for static data, ESA PECS project NaturaSat Deliverable 2.1, 2018.
[16] G. Nakamura and R. Potthast, *Inverse Modeling*, IOP Expanding Physics, IOP Publishing, Bristol, 2015, 2053–2563.
[17] P. Pauš, M. Beneš, Algorithm for topological changes of parametrically described curves, *Proceedings of ALGORITMY*, 2009, 176–184.
[18] A. Sarti, R. Malladi and J. A. Sethian: Subjective surfaces: A method for completing missing boundaries, *Proc. Natl. Acad. Sci. USA*, **97** (2000), 6258–6263.

Received December 2018; revised November 2019.

E-mail address: mikula@math.sk
E-mail address: jozo.urban@gmail.com
E-mail address: michalkollar27@gmail.com
E-mail address: ambroz.martin.ml@gmail.com
E-mail address: ivan.jarolimek@savba.sk
E-mail address: jozef.sibik@savba.sk
E-mail address: maria.sibikova@savba.sk