DISCRETE AND CONTINUOUS
DYNAMICAL SYSTEMS SERIES S

doi:10.3934/dcdss.2020231

SEMI-AUTOMATIC SEGMENTATION OF NATURA 2000 HABITATS IN SENTINEL-2 SATELLITE IMAGES BY EVOLVING OPEN CURVES

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

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

Ivan Jarolímek, Jozef Šibík and Mária Šibíková

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

Abstract. In this paper we introduce mathematical model and real-time numerical method for segmentation of Natura 2000 habitats in satellite images by evolving open planar curves. These curves in the Lagrangian formulation are driven by a suitable velocity vector field, projected to the curve normal. Besides the vector field, the evolving curve is influenced also by the local curvature representing a smoothing term. The model is numerically solved using the flowing finite volume method discretizing the arising intrinsic partial differential equation with Dirichlet boundary conditions. The time discretization is chosen as an explicit due to the ability of real-time edge tracking. We present the results of semi-automatic segmentation of various areas across Slovakia, from the riparian forests to mountainous areas with scrub pine. The numerical results were compared to habitat boundaries tracked by GPS device in the field by using the mean and maximal Hausdorff distances as criterion.

1. Introduction.

In this paper we present a semi-automatic method for image segmentation of NATURA 2000 habitats in Sentinel-2 satellite images. In the section 2 the main idea of our segmentation method is given. We design the velocity vector field and we prescribe the evolution equation for the segmentation curve in the form of an intrinsic partial differential equation with Dirichlet boundary conditions. Similar models are used in the context of dislocation dynamics in [5, 10]. Section 3 deals with the numerical discretization of the proposed segmentation model. The comparison of numerically segmented areas and areas tracked by GPS device is presented in section 4.

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 and ESA Contract No. 4000122575/17/NL/SC.

2. Mathematical model for semi-automatic image segmentation.

In this section we explain design of the segmentation model by using artificial images. We use a suitable velocity vector field, based on the smoothed image information, which drives the evolving curve automatically to the boundary of segmented habitat. Such vector field is constructed by using the pre-smoothed image intensity gradient as an edge indicator which is an input of the edge detector function. Then the gradient of the edge detector function is projected to the normal of the curve and the overall curve motion is regularized by using the local curvature of evolving curve [3, 4, 7, 8]. The final mathematical model is given by the corresponding nonlinear intrinsic partial differential equation which is discretized and solved numerically by the flowing finite volume method [9, 7, 1].

2.1. Construction of the velocity vector field.

Let us consider a 2D scalar function of image intensity $I^0 : \mathbb{R}^2 \to \mathbb{R}$, given e.g. by a simple artificial example plotted in Fig. 1 left, and also the function I^σ representing a Gaussian smoothing of the original image, see Fig. 1 right.

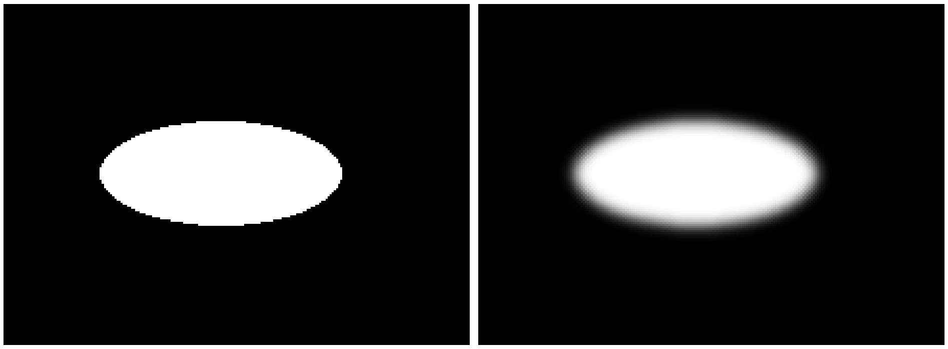

FIGURE 1. An example of artificial greyscale 2D image (left) and smoothed image (right).

In other words, $I^0(\mathbf{x}) = I^0(x_1, x_2)$ represents a scalar value of the image intensity in a particular (pixel) coordinates x_1 and x_2. In case of the Sentinel-2 optical data we work, in general, with a combination of three chosen optical bands, so we consider the image intensity as a vector function $I^0 : \mathbb{R}^2 \to \mathbb{R}^3$. However, the main ideas of our semi-automatic segmentation approach can be simply explained in the scalar case, so we do it in that way and emphasize arising differences only when necessary.

The smoothed greyscale image intensity function I^σ for the image in Fig. 1 can be also represented by a 3D graph as plotted in Fig. 2 left. The boundaries of segmented areas are usually represented by edges in the image and thus the edges are the most important features which should attract the evolving segmentation curve. The image edges can be identified by sharp changes of the smoothed image intensity which is mathematically characterized by a large value of the norm of image intensity gradient $|\nabla I^\sigma(\mathbf{x})|$. In Fig. 2 right one can see a 3D graph of $|\nabla I^\sigma(\mathbf{x})|$ calculated for our illustrative image from Fig. 1 right and its image intensity function plotted in Fig. 2 left.

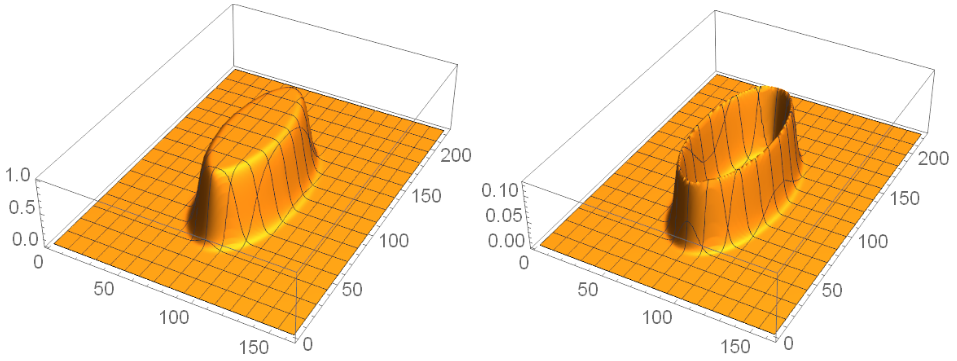

FIGURE 2. 3D graph of image intensity function $I^\sigma(\mathbf{x})$ (left) and graph of the image intensity gradient norm $|\nabla I^\sigma(\mathbf{x})|$ (right), for the smoothed image in Fig. 1 right.

The norm of gradient $|\nabla I^\sigma(\mathbf{x})|$ forms an input to an edge detector function g defined by [11]

$$g(k, |\nabla I^\sigma(\mathbf{x})|) = \frac{1}{\left(1 + k\,|\nabla I^\sigma(\mathbf{x})|^2\right)}, \tag{1}$$

where k is a so-called scaling factor. The values of $g(k, |\nabla I^\sigma(\mathbf{x})|)$ are small along the image edges and large elsewhere, see Fig. 3.

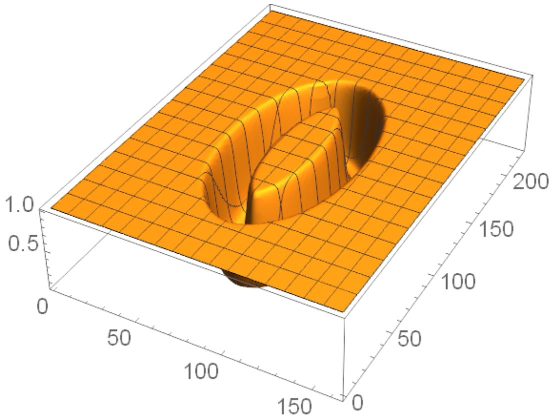

FIGURE 3. 3D graph of the edge detector $g(|\nabla I^\sigma(\mathbf{x})|)$ for the smoothed image in Fig. 1 right.

Inspired by [3, 4] we can define a velocity vector field $\mathbf{v}(\mathbf{x})$ by taking the gradient of the edge detector function with minus sign,

$$\mathbf{v}(\mathbf{x}) = -\nabla g(|\nabla I^\sigma(\mathbf{x})|), \tag{2}$$

and see in Fig. 4 that such vector field points always towards the edges in the image and thus can be used as a force driving segmentation curve always in a correct direction, to the habitat border lines, see also Fig. 11 in section 4.

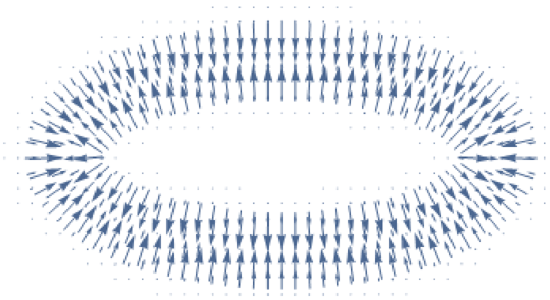

FIGURE 4. A visualization of the vector field $\mathbf{v}(\mathbf{x})$ for image in Fig. 1 right. We see that arrows points to the edge in the image from both sides.

2.2. **The curve evolving in vector field.** Let Γ be an open planar curve, $\Gamma : [0,1] \to \mathbb{R}^2$, $\Gamma = \{\mathbf{x}(t,u), u \in [0,1]\}$, depending on time t and let $\mathbf{x}(t,u) = (\mathbf{x}_1(t,u), \mathbf{x}_2(t,u))$ be a position vector of the curve Γ for parameter u in time t. In the sequel, the curve will be discretized and represented by a set of points. An example of an open planar curve discretization is displayed in Fig. 5, where $\mathbf{x}_0^m, \mathbf{x}_1^m, ..., \mathbf{x}_n^m$ are discrete curve points which correspond to the uniform discretization of the interval $[0,1]$ with spatial step $h = 1/n$ at m-th time with time step τ. Due to the Dirichlet boundary conditions we have that $\mathbf{x}(t,0) = \mathbf{x}(0,0)$ and $\mathbf{x}(t,1) = \mathbf{x}(0,1)$, $t > 0$.

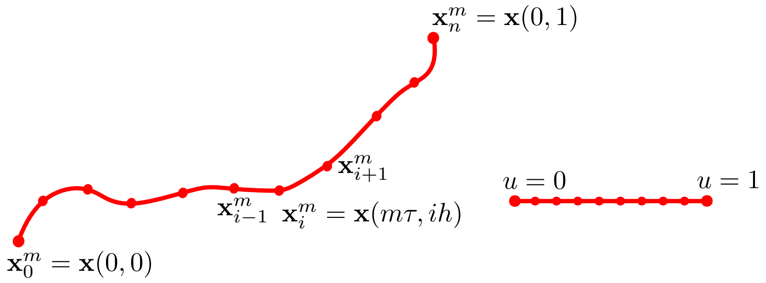
$\mathbf{x}_n^m = \mathbf{x}(0,1)$
$\mathbf{x}_{i+1}^m$
$u = 0$
$u = 1$
$\mathbf{x}_{i-1}^m$
$\mathbf{x}_i^m = \mathbf{x}(m\tau, ih)$
$\mathbf{x}_0^m = \mathbf{x}(0,0)$

FIGURE 5. An open curve discretization (left) corresponding to the uniform discretization of parameter $u \in [0,1]$ (right).

2.2.1. *Evolution driven by the vector field.* The curve evolution driven by the vector field $\mathbf{v}(\mathbf{x})$ is given by a nonstationary differential equation

$$\mathbf{x}_t = \mathbf{v}(\mathbf{x}), \tag{3}$$

where $\mathbf{x}_t$ denotes a partial derivative of $\mathbf{x}$ with respect to t. In our approach the segmentation curve evolution begins from a uniformly distributed line segment, defined by a user e.g. by mouse clicks, which is then moved automatically towards the edge in the image, see Fig. 6 for the discrete curve point evolution to the edge in the artificial image from Fig. 1.

We can clearly see that such curve evolution brings suitable results for simple images, however, it can be fairly inaccurate in more complicated cases and it needs

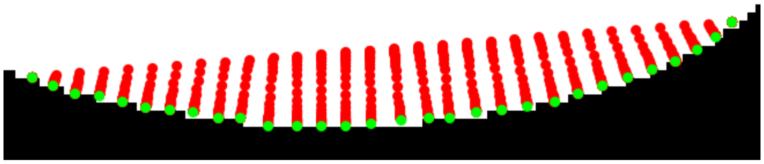

FIGURE 6. Trajectories of points of a discrete segmentation curve (red) evolved in the vector field **v** driven to the image edge. The final state of discrete segmentation curve is given by green points localized on the image edge.

important modifications. Due to the discrete character of the vector field and evolving curve as well, one of the problems arising is an accumulation of a discrete curve points during the curve evolution and in the steady state. This phenomenon is documented in Fig. 7, where more complicated objects were segmented by the above simple approach, and lead us to a modification of the basic model (3).

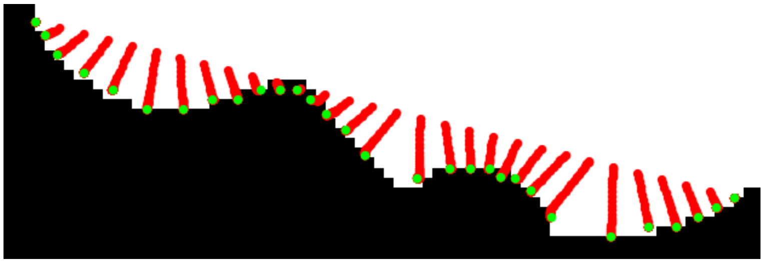

FIGURE 7. Trajectories of points of a discrete segmentation curve (red) evolved in the vector field **v** and their final position (green) visualized over the original image. One can see a problem of non-uniform distribution of points on evolving discrete segmentation curve due to non-controlled tangential velocities in the vector field **v**.

2.2.2. *Ignoring of the uncontrolled tangential velocity component.* In general, we can describe the curve evolution given by Eq. (3) as a motion in normal and tangential directions

$$\mathbf{x}_t = \beta \mathbf{N} + \alpha \mathbf{T}, \tag{4}$$

where α is a tangential velocity, $\mathbf{T} = \mathbf{x}_s$ is a unit tangent vector with s being the arc-length parametrization of the curve $\Gamma, ds = G du$, $G = |\mathbf{x}_u|$, β is a normal velocity and $\mathbf{N} = \mathbf{T}^{\perp} = ((x_2)_s, -(x_1)_s)$ is a unit normal vector.

Although the tangential component of velocity vector does not change the overall curve shape, it only reparametrizes the curve and moves the points along the curve in tangential direction, it can cause a non-uniform distribution of points on the curve, see Fig. 7, and thus serious numerical errors in realistic situations. If we ignore it, we can eliminate such undesired movements. Therefore, if we start from uniformly distributed initial abscissa and move it only in normal direction, the evolving curve will remain almost uniformly discretized, see Fig. 8. In such case,

$\alpha = 0$ in Eq. (4), and we can rewrite the curve evolution equation into the following form

$$\mathbf{x}_t = \lambda v_{\mathbf{N}} \mathbf{N}, \tag{5}$$

where $\lambda > 0$ is a parameter and the nonlinear term $v_{\mathbf{N}}$ represents the projection of velocity vector $\mathbf{v}$ onto the normal $\mathbf{N}$ of the moving curve,

$$v_{\mathbf{N}} = \mathbf{v} \cdot \mathbf{N}. \tag{6}$$

Removing the non-controlled tangential part of velocity causes better distribution of curve grid points as can be seen in Fig. 8.

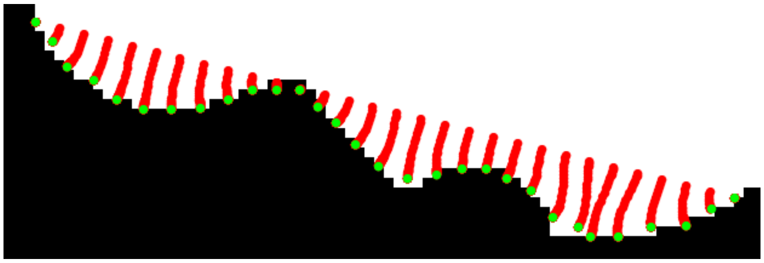

FIGURE 8. Trajectories of points of a discrete segmentation curve (red) evolved in the vector field $\mathbf{v}$ and their final position (green) visualized over the original image. An improved distribution of the curve grid points after removing the tangential component of the velocity vector field $\mathbf{v}$ is obvious.

2.2.3. *The regularization by curvature.* The mathematical model (5) can be even more improved and numerically stabilized by incorporating the local curvature information. Incorporating the curvature yields a sensitivity of the numerical discretization to a distance of neighbouring points, which ties together discrete points of the evolving curve. In Fig. 9 we illustrate a situation, when the image is less smoothed and thus the velocity vector field is almost zero in some points of initial abscissa. Clearly, by using numerically discretized equation (5), these points cannot move, normal direction is changed and the evolution is not as one desires, see Fig. 9 top. On the other hand, incorporating the local curvature influence, we smooth the evolution, ties all points together in numerical discretization and get the result plotted in Fig. 9 bottom. So we modify the curve evolution equation (5) into the following form

$$\mathbf{x}_t = \lambda v_{\mathbf{N}} \mathbf{N} + \delta k \mathbf{N}, \tag{7}$$

where δ is a parameter and the term $k\mathbf{N}$ represents the so-called curvature vector. From the Frenet equations we get for the curvature vector identities $k\mathbf{N} = \mathbf{T}_s = \mathbf{x}_{ss}$. Using this we obtain final intrinsic nonlinear partial differential equation for evolution of the segmentation curve

$$\mathbf{x}_t = \lambda v_{\mathbf{N}} \mathbf{N} + \delta \mathbf{x}_{ss}. \tag{8}$$

The parameters λ and δ weight the vector field influence and the curvature influence.

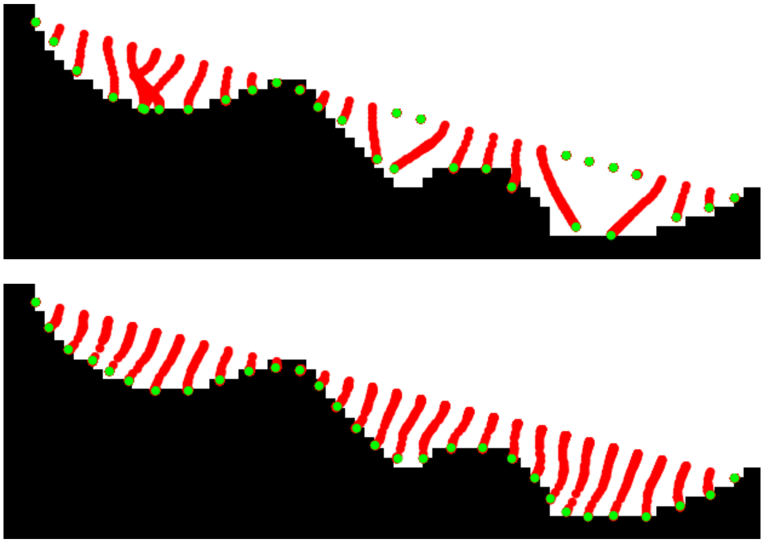

FIGURE 9. Trajectories of points of a discrete segmentation curve (red) evolved in the vector field $\mathbf{v}$ and their final position (green) visualized over the original image. Top: the curve evolution by (5) in a less smoothed image when problem of crossing, accumulating and not moving points may arise; bottom: such undesired behaviour is removed by employing the local curvature influence into the model (7).

3. **Numerical discretization.** Let us recall the intrinsic PDE (8) for the open curve evolution and write it in the following form suitable for numerical discretization

$$\mathbf{x}_t = \delta \mathbf{x}_{ss} + w \mathbf{x}_s^{\perp}, \tag{9}$$

where $w = \lambda v_{\mathbf{N}}$. First, we perform the spatial discretization, which is based on the flowing finite volume method [9, 2, 1].

Integrating (9) over the finite volume $\mathbf{p}_i = [\mathbf{x}_{i-\frac{1}{2}}, \mathbf{x}_{i+\frac{1}{2}}]$, see Fig. 10, where $\mathbf{x}_{i-\frac{1}{2}}$ represents the middle point between the points $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ and ds is an integration element of piecewise linear approximation of original curve, i.e.

$$\mathbf{x}_{i-\frac{1}{2}} = \frac{\mathbf{x}_{i-1} + \mathbf{x}_i}{2}, \quad i = 1, \ldots, n \tag{10}$$

we get

$$\int\limits_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \mathbf{x}_t ds = \delta \int\limits_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \mathbf{x}_{ss} ds + w \int\limits_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \mathbf{x}_s^{\perp} ds, \tag{11}$$

where the values δ and w are considered constant, with values δ_i and 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 $\frac{h_i + h_{i+1}}{2}$. Using the Newton-Leibniz formula, (10) and using approximation of the arc-length derivative $\mathbf{x}_s$ by a finite difference

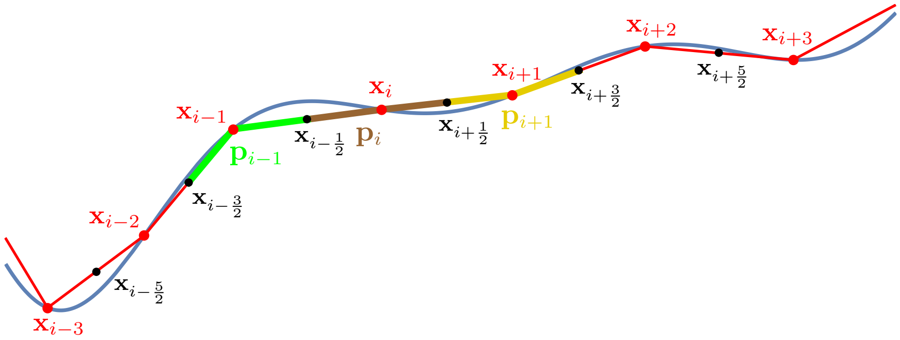

FIGURE 10. Visualization of the curve discretization [1] curve grid points (red), discrete curve segments (different colors) and the midpoints (black). Finite volumes $\mathbf{p}_{i-1}, \mathbf{p}_i$, and $\mathbf{p}_{i+1}$ are highlighted by green, brown and yellow color. Note that $\mathbf{p}_i$ is 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 $\mathbf{x}_{i+\frac{1}{2}}$.

we get semi-discrete flowing finite volume scheme

$$\frac{h_i + h_{i+1}}{2} \left(\mathbf{x}_i\right)_t = \delta_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h_{i+1}} - \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{h_i} \right) + w_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_{i-1}}{2} \right)^{\perp}, \quad (12)$$

for $i = 1, ..., n-1$. 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 $\left(\mathbf{x}_i\right)_t = \frac{\mathbf{x}_i^{m+1} - \mathbf{x}_i^m}{\tau}$. Approximating both the vector field term and the curvature term explicitly we obtain the fully discrete explicit scheme

$$\begin{aligned} \mathbf{x}_i^{m+1} = \mathbf{x}_i^m + \tau \delta_i^m \frac{2}{h_{i+1}^m + h_i^m} \left(\frac{\mathbf{x}_{i+1}^m - \mathbf{x}_i^m}{h_{i+1}^m} - \frac{\mathbf{x}_i^m - \mathbf{x}_{i-1}^m}{h_i^m} \right) \\ + \tau w_i^m \frac{\left(\mathbf{x}_{i+1}^m - \mathbf{x}_{i-1}^m\right)^{\perp}}{h_{i+1}^m + h_i^m} \end{aligned} \quad (13)$$

for $i = 1, ..., n-1$, where n is the number of the curve grid points. Due to boundary conditions we have $\mathbf{x}_0^m = \mathbf{x}_0^0$ and $\mathbf{x}_n^m = \mathbf{x}_n^0$. The parameter w_i^m is given as follows

$$w_i^m = \lambda v_{\mathbf{N}_i}^{\ m} = \lambda \mathbf{v}(\mathbf{x}_i^m) \cdot \mathbf{N}_i^m = \lambda \mathbf{v}(\mathbf{x}_i^m). \frac{\left(\mathbf{x}_{i+1}^m - \mathbf{x}_{i-1}^m\right)^{\perp}}{h_{i+1}^m + h_i^m}. \quad (14)$$

4. **Numerical experiments.** In the first numerical experiment, we show behaviour of the model (7) in real data. As one can see in Fig. 11, the model (7) and its numerical discretization (13) cause a regular, almost uniform distribution of grid points during the evolution and in the segmentation result. Moreover, the numerical scheme (13) can be efficiently implemented and allows real-time segmentation of habitats.

We note here that in the segmentation of real data we use three optical channels and thus we consider the vector image intensity function I^0 defined at the beginning of section 2.1. However, the only change in the mathematical model (7) and its numerical discretization (13) is that, instead of the norm of gradient of the scalar image intensity I^σ in Eq. (2) we consider the average of norms of gradients of image intensities in all three channels. Moreover, in our implementation, the Gaussian convolution is realized numerically by solving the linear heat equation in one discrete time step using the implicit scheme. The coefficients k, λ and δ are chosen by the user such that the semi-automatic method would give desired results. These coefficients can be changed during the segmentation, which is natural for such real time method.

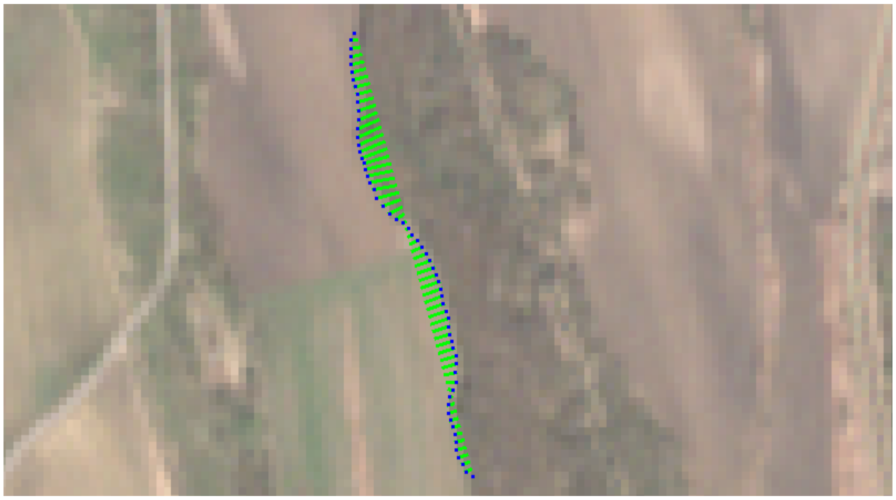

FIGURE 11. A discrete segmentation curve evolving to habitat boundary in a real 3-band Sentinel-2 optical image. The green color shows trajectories of moving discrete curve points and blue points represents the result of segmentation of this particular section of the habitat border.

In Fig. 11 we present just one section of the segmentation curve and it illustrates the procedure how a user performs semi-automatic segmentation. The user clicks the mouse at some correctly chosen point on the habitat boundary and drag the mouse along the expected habitat boundary. The algorithm always connects the first clicked point with the last mouse position, constructs the straight line between them and in real-time adjusts this line to the habitat border by using the numerical scheme (13). When the user is satisfied with the detected borderline, clicks the mouse again and that portion of segmentation is finished. By repeating the procedure the user can continue along the borderline and eventually he clicks on the first point of the first segment to close the boundary curve. We illustrate this consecutive procedure in Fig. 12.

4.1. **Comparison of discrete curves.** After performing the semi-automatic segmentation of habitats in Sentinel-2 data, we compare the segmentation results with the GPS tracks obtained by botanists in the field. For this quantitative comparison of two curves we use the classical (maximal) Hausdorff distance [12] and the so-called mean Hausdorff distance, see e.g. [6], which are general tools for computing distance of curves, surfaces and even more complicated geometrical continuous or discrete objects (sets). The mean Hausdorff distance $\overline{\mathrm{d}}_H(A, B)$ is given by the

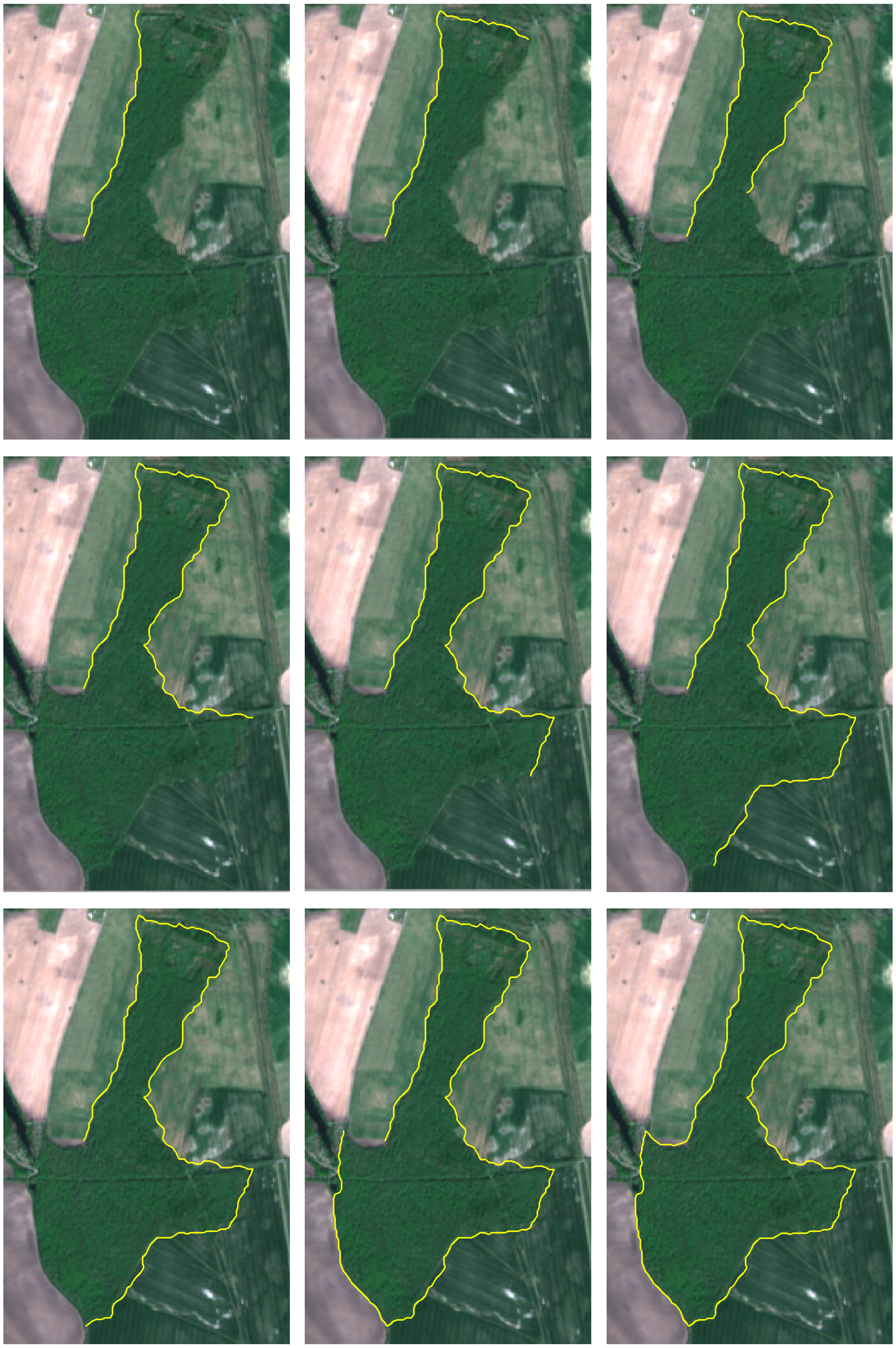

Figure 12. An example of the semi-automatic segmentation showing consecutive building of the segmentation curve (yellow), the final result is on the bottom right.

following formulae

$$\overline{\delta}_H(A,B) = \frac{1}{n}\sum_{i=1}^{n}\min_{b\in B} d\left(a_i,b\right), \overline{\delta}_H(B,A) = \frac{1}{m}\sum_{i=1}^{m}\min_{a\in A} d\left(a,b_i\right), \tag{15}$$

$$\overline{\mathrm{d}}_H(A,B) = \frac{\overline{\delta}_H(A,B)+\overline{\delta}_H(B,A)}{2}, \tag{16}$$

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

$$\mathrm{d}_H(A,B) = \max\left\{\sup_{a\in A}\inf_{b\in B}\mathrm{d}(a,b), \sup_{b\in B}\inf_{a\in A}\mathrm{d}(a,b)\right\}. \tag{17}$$

In order to test reliability and usability of the semi-automatic segmentation method, together 24 areas of riparian forests - Natura 2000 habitat *91F0 Riparian mixed forests along the great river* - were tracked by GPS device. Due to the variable borders, they are suitable for testing the ability and performance of developed semi-automatic segmentation tool to detect accurately their shape. In case of problems in the field, mainly due to flooded parts of forests where it was impossible to walk around, some GPS tracks were corrected in the Google Earth Pro software. The mean Hausdorff distance is in average 11.48 m which is very close to the pixel resolution (10m) of Sentinel-2 data. This means that by the semi-automatic segmentation we are able to detect habitat borders as accurately as the image resolution allows. Maximal Hausdorff distance is in average about 58m, what represents 5-6 pixels. Some differences can be found only in areas with cotone zones where tree dominated riparian forests are connected to surrounding meadows or fields by shrub dominated zone, see Figs. 13 and 14.

Next, 18 areas of Natura 2000 habitat *4070* Bushes with Pinus mugo and Rhododendron hirsutum (Mugo-Rhododendretum hirsuti)* were tracked in various mountain ranges in Slovakia (Malá Fatra Mts., Západné Tatry Mts., Nízke Tatry Mts., Chočské vrchy Mts. and Oravské Beskydy Mts.). Bushes with Pinus mugo usually form large areas of diversified shape, discontinued by avalanche gullies, small mountain creeks or by glacially formed moraines. Considering this fact, the correct semi-automatic segmentation is a challenging task. Moreover, the field mapping of habitat borders in high-altitude rugged terrain is very complicated and time-consuming, so using the satellite image segmentation methods seem to be very efficient and promising way of monitoring this habitat. In general, we observed that the mean Hausdorff distances of GPS tracked and semi-automatically segmented borders of bushes with Pinus mugo areas are also close to the pixel resolution of Sentinel-2 data, 13.9m in average of all 18 areas. The maximal Hausdorff distances are in general bigger than those observed in the areas of riparian forests. Bushes with Pinus mugo grow on large areas connected with the mountain spruce forests. Some extreme values of the maximal Hausdorff distance are caused by the "ecotone zone" where botanist in the field takes subjective decision about the habitat border, for illustration of such problematic areas see Figs. 15 and 16.

5. Conclusions.

In this paper we proposed a semi-automatic segmentation method of Natura 2000 habitats in Sentinel-2 optical data. We discussed the segmentation curve velocity vector field design and described the curve evolution numerical algorithm using the explicit scheme allowing real-time boundary tracking. We also

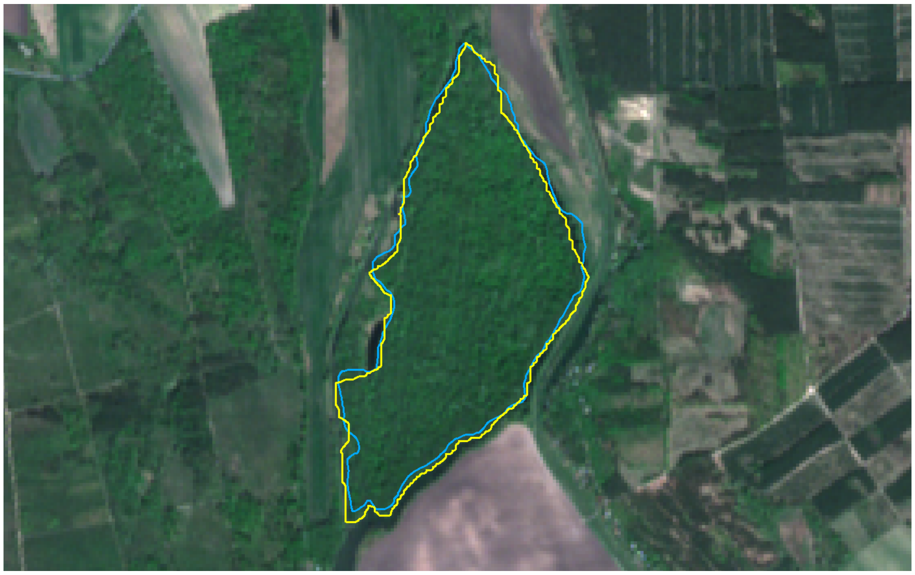

FIGURE 13. Semi-automatic segmentation (yellow) and GPS track (light-blue) with almost exact overlap. The maximal Haussdorff distance is 62.1m and the mean Hausdorff distance is 14.0m in this case, which means that we obtained almost the pixel resolution (10m) accuracy.

presented numerical experiments using Sentinel-2 optical images. The comparison of areas obtained by the semi-automatic segmentation and by the GPS tracking in the field shows that we can get the accuracy compared to the pixel resolution of Sentinel-2 data. Some further improvements of the numerical model, e.g. asymptotically uniform curve grid point redistribution, will be treated in the future together with the implementation of the semi-implicit scheme. Here the real-time performance of the method will be also a discretization quality criterion.

REFERENCES

[1] M. Ambroz, M. Balažovjech, M. Medla and K. Mikula, Numerical modeling of wildland surface fire propagation by evolving surface curves, *Advances in Computational Mathematics*, **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, *Proceedings of ALGORITMY*, (2012), 42–52.

[3] V. Caselles, R. Kimmel and G. Sapiro, Geodesic active contours, *International Journal of Computer Vision*, **22** (1997), 61–79.

[4] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum and A. Yezzi, Jr., Conformal curvature flows: From phase transitions to active vision, *Arch. Rational Mech. Anal.*, **134** (1996), 275–301.

[5] M. Kolář, M. Beneš and D. Ševčovič, Computational analysis of the conserved curvature driven flow for open curves in the plane, *Mathematics and Computers in Simulation*, **126** (2016), 1–13.

[6] Z. Krivá, K. Mikula, M. Peyriéras, B. Rizzi, A. Sarti and O. Stašová , 3D early embryogenesis image filtering by nonlinear partial differential equations, *Medical Image Analysis*, **14** (2010), 510–526.

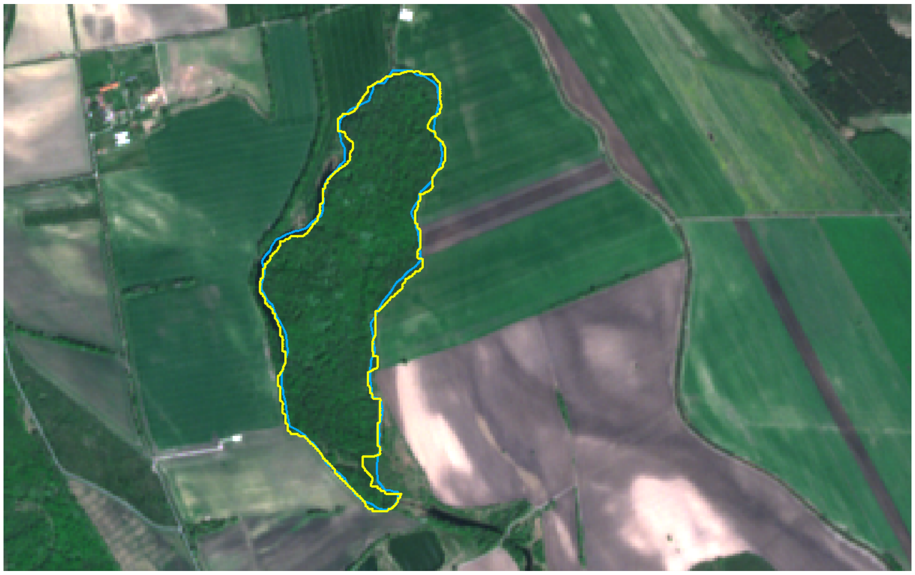

FIGURE 14. An example of a complicated border of the riparian forest. We compare the semi-automatic segmentation (yellow) and GPS track (light-blue). The mean Haussdorff distance is 12.0m and the maximal Haussdorff distance, in this case, is 62.1m, indicating correctly discrepancy in habitat area estimate in the field and by employing the Sentinel-2 optical data.

[7] K. Mikula and D. Ševčovič, Computational and qualitative aspects of evolution of curves driven by curvature and external force, *Computing and Visualization in Science*, **6** (2004), 211–225.

[8] K. Mikula and D. Ševčovič, Evolution of curves on a surface driven by the geodesic curvature and external force, *Applicable Analysis*, **85** (2006), 345–362.

[9] K. Mikula, D. Ševčovič and M. Balažovjech, A simple, fast and stabilized flowing finite volume method for solving general curve evolution equations, *Communications in Computational Physics*, **7** (2010), 195–211.

[10] P. Pauš, M. Beneš, M. Kolář and J. Kratochvíl, Dynamics of dislocations described as evolving curves interacting with obstacles, *Modelling and Simulation in Materials Science and Engineering*, **24** (2016), 34 pp.

[11] P. Perona and J. Malik, Scale space and edge detection using anisotropic diffusion, *Proceedings of the IEEE Society Workshop on Computer Vision*, **12** (1987), 629–639.

[12] Hausdorff distance, 20 12 2018, https://en.wikipedia.org/wiki/Hausdorff_distance.

Received January 2019; revised September 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

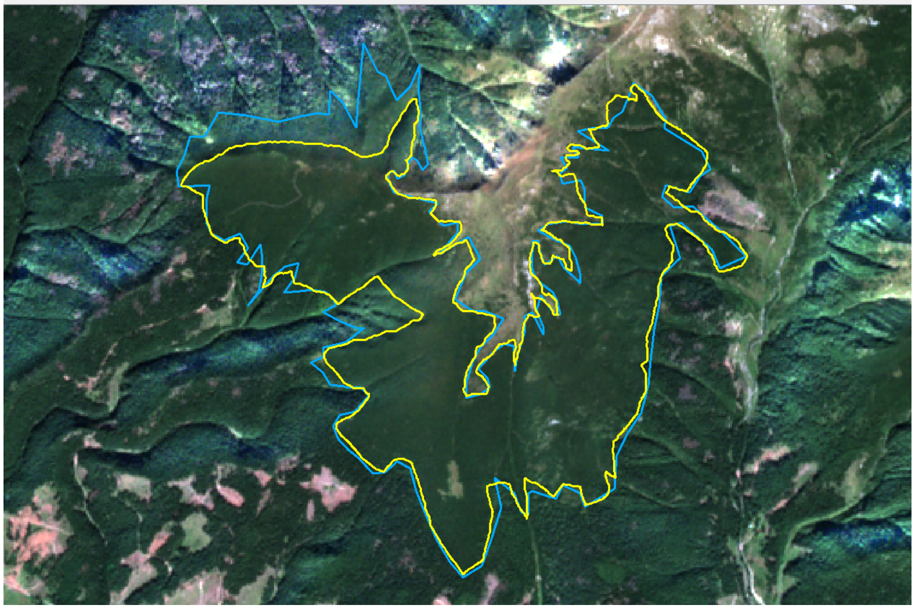

Figure 15. The locality with the highest, 413.3m, maximal Hausdorff distance between semi-automatically segmented and GPS tracked curves among bushes with Pinus mugo tested areas, here also the mean Hausdorff distance was the highest, 44.8m. On the North-West habitat border, we can see the "ecotone zone" that was included during field tracking (light-blue) and excluded by using the semi-automatic segmentation (yellow).

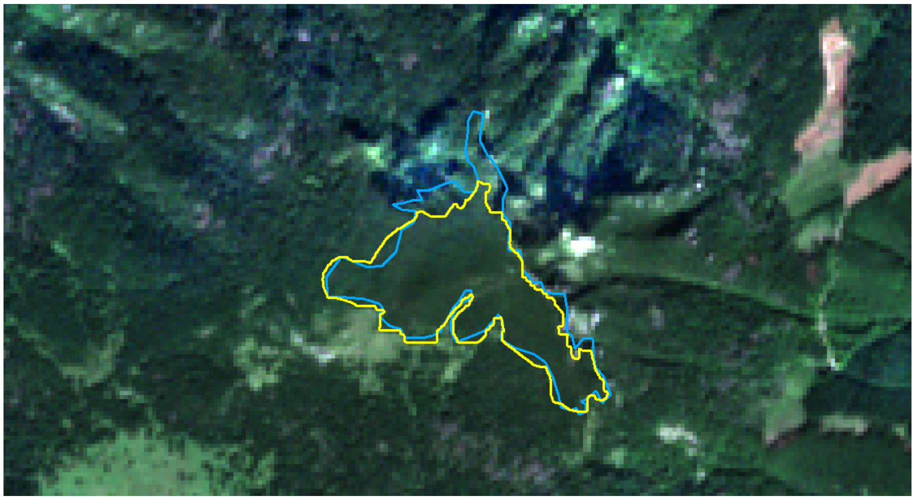

Figure 16. The locality dominated by Pinus mugo with the "ecotone zone" that was included during the field tracking (light-blue) and excluded by using the semi-automatic segmentation (yellow). The mean Hausdorff distance is 19.1m and the maximal Hausdorff distance is 171.0m.