

SEMI-IMPLICIT SCHEME FOR SEMI-AUTOMATIC SEGMENTATION IN NATURASAT SOFTWARE *

MARTIN AMBROZ [†], MICHAL KOLLÁR [‡], AND KAROL MIKULA [§]

Abstract. In this paper, we present new accurate and efficient semi-automatic segmentation method for Sentinel-2 satellite images which is used in NaturaSat software for exploration of Natura 2000 habitats. The method is based on the Lagrangian approach to evolving plane curves and the mathematical model incorporates the tangential redistribution of grid points. The Lagrangian curve evolution is solved by the semi-implicit scheme and new robust and efficient initial condition for curve evolution is proposed. The final semi-automatic segmentation is automatically adjusted to represent the desired boundaries accurately with a control of the smoothness of the final segmentation result. Together with comprehensive discussion on semi-automatic segmentation, the NaturaSat software design and its functionality is presented as well.

Key words. Image segmentation, curve evolution, numerical method, Natura 2000, satellite images, Sentinel-2.

AMS subject classifications. 35R01, 65M08, 35R37, 92F05.

1. Introduction. This paper presents new accurate and efficient semi-automatic method for 2D satellite image segmentation. It is based on the Lagrangian approach to evolving plane curves, see e.g. [16, 17]. Comparing to the paper [16] it brings *(i)* new mathematical model including tangential redistribution of points, *(ii)* the Lagrangian curve evolution is solved not by an explicit but by the semi-implicit scheme and *(iii)* new robust and efficient initial condition for the curve evolution is proposed. Moreover, *(iv)* at the final step of the semi-automatic segmentation curve is automatically adjusted to represent the Natura 2000 habitat boundary as accurately as possible with a control of the smoothness of the final segmentation result.

By its name, the semi-automatic segmentation requires a user interaction. In our approach the user performs the segmentation in several subsequent steps and the overall segmentation result is given by an interconnection of several open curve segments. The quality of these segments is controlled by the user and governed by the mathematical model and its parameters. In Fig. 1.1 we illustrate the subsequent building of the overall segmentation curve (yellow) by the semi-automatic approach.

The outline of the paper is as follows. First, we focus on the mathematical model used. Then we present numerical algorithms including construction of the initial condition, derivation of the semi-implicit numerical scheme and the final automatic segmentation curve adjustment. At the same time, in Figures we illustrate the behaviour of developed algorithms and show basic features the software NaturaSat which was built for the Sentinel-2 satellite image segmentation in order to explore Natura 2000 habitats areas and quality.

*This work was supported by grants APVV-16-0431, APVV-19-0460, VEGA 1/0436/20.

[†]Department of Mathematics, Slovak University of Technology, Radlinskeho 11, 81005 Bratislava, Slovakia (martin.ambroz@stuba.sk)

[‡]Department of Mathematics, Slovak University of Technology, Radlinskeho 11, 81005 Bratislava, Slovakia (kollarm@math.sk)

[§]Department of Mathematics, Slovak University of Technology, Radlinskeho 11, 81005 Bratislava, Slovakia and Algoritmy:SK, s.r.o., Bratislava, Slovakia (mikula@math.sk).

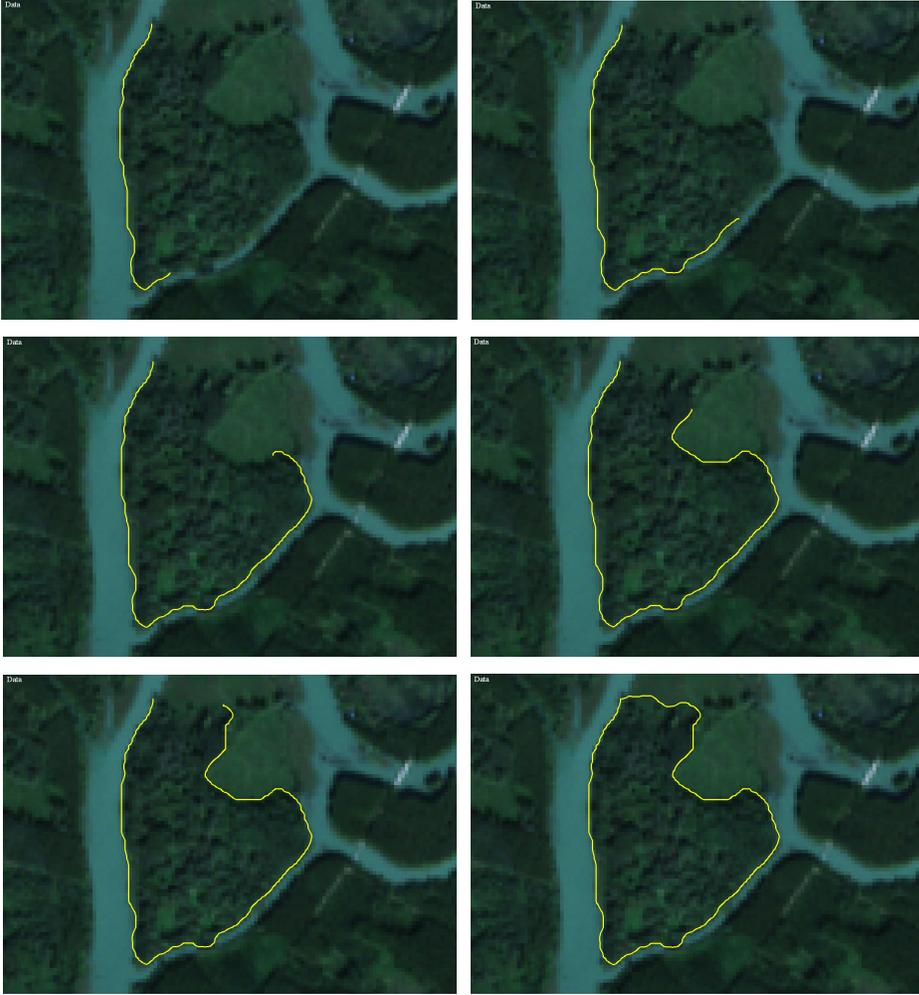


FIG. 1.1. *The semi-automatic segmentation done in 6 subsequent steps.*

2. Semi-automatic segmentation method. The first semi-automatic segmentation algorithm for the Sentinel-2 satellite data was described in [16] where the segmentation curve was driven by the equation

$$\mathbf{x}_t = \lambda v_{\mathbf{N}} \mathbf{N} + \delta k \mathbf{N}, \quad (2.1)$$

where $\lambda > 0$, $\delta > 0$ are model parameters,

$$v_{\mathbf{N}} = \mathbf{v} \cdot \mathbf{N} \quad (2.2)$$

represents the projection of velocity vector \mathbf{v} onto the normal \mathbf{N} of the moving segmentation curve and $k\mathbf{N}$ represents its curvature vector.

The velocity vector field \mathbf{v} in (2.2) is given by the gradient of the edge detector function g with the minus sign, i.e.,

$$\mathbf{v}(\mathbf{x}) = -\nabla g(|\nabla I^\sigma(\mathbf{x})|) \quad (2.3)$$

where the edge detector function is given by

$$g(\mathbf{x}) = \frac{1}{1 + k|\nabla I^\sigma(\mathbf{x})|^2} \quad (2.4)$$

and $|\nabla I^\sigma(\mathbf{x})| = |\nabla G_{\sigma_0} * I^0|(\mathbf{x})$, I^σ is a Gaussian smoothing of the original image I^0 , $k \geq 0$.

In [16] it was also shown that the tangential component of the velocity vector field \mathbf{v} is not necessarily useful and the model has better results when omitting it, i.e. to use just $v_{\mathbf{N}}\mathbf{N}$ as the curve driving force in (2.1). On the other hand, using a suitable tangential velocity, which redistributes the curve grid points asymptotically uniformly, could improve the segmentation results. Such approach to tangential velocity, improving mainly the robustness and stability of numerical computations, was successfully used in a number of applications [7, 9, 1] including the automatic segmentation of satellite images [17]. By incorporating the tangential velocity α into the semi-automatic segmentation we get the mathematical model in the following form

$$\mathbf{x}_t = \lambda v_{\mathbf{N}}\mathbf{N} + \delta k\mathbf{N} + \alpha\mathbf{T}, \quad (2.5)$$

where $\mathbf{T} = \mathbf{x}_s = ((x_1)_s, (x_2)_s)$ is the unit tangent vector of the curve with s being the arc-length parametrization of the curve, and $\mathbf{N} = \mathbf{T}^\perp = (-(x_2)_s, (x_1)_s)$ is a unit normal vector. The model (2.5) can be written in the following form

$$\mathbf{x}_t = \beta\mathbf{N} + \alpha\mathbf{T}, \quad (2.6)$$

where $\beta = \lambda v_{\mathbf{N}} + \delta k$ is the normal velocity. The suitable (asymptotically uniform) tangential velocity is given by [13]

$$\alpha_s = k\beta - \langle k\beta \rangle_\Gamma + \omega \left(\frac{L}{g} - 1 \right) \quad (2.7)$$

where k is the curvature, β the normal velocity, g the local length, L the global length of the curve, ω the redistribution parameter, $\langle k\beta \rangle_\Gamma$ the average of $k\beta$ along the curve and α_s is the derivative of α with respect to the arclength parameter s .



FIG. 2.1. *Left: segmentation by the abscissa initial condition with long distance end-points, Right: segmentation by using new initial condition with long distance end-points. The yellow curves represent the initial conditions and red curves the segmentation results.*

2.1. The initial condition. In [16], the initial condition was given by an abscissa connecting two points given by mouse clicks of the user. The abscissa is then adjusted by the model (2.1) to the image edge and creates the first segmentation curve segment. Its last point becomes the first point of initial abscissa for the next segment and the user clicks by mouse the second end point and the process is repeated. Such approach works correctly if subsequent segments are not too long, otherwise, it is not easy for the curve evolution model to find reliably the correct boundary of the segmented area, see Fig. 2.1 left where the initial curve (yellow) is attracted to the nearest edges and not to a desired one. In this paper, we propose a different algorithm for the initial condition construction, which significantly improves the semi-automatic segmentation speed and also the quality of resulting segmentation. The basic idea is to roughly approximate a suitable iso-line of the edge detector function g , which is a potential for the vector field \mathbf{v} , already in the construction of the initial condition. The resulting initial curve is already close to the desired edge on a long distance between two clicked points and it is then attracted fast to the boundary of the segmented area.

The inputs to the algorithm are:

- the initial point \mathbf{x}_1 ,
- the final point \mathbf{x}_F ,
- the vector field \mathbf{v} and a desired curve discretization step length h_d .

Setting $i = 1$ we repeat until $|\mathbf{x}_F - \mathbf{x}_i| < h_d$:

1. compute $\mathbf{v}^\perp = (-v_2, v_1)$ as the normal vector to the vector field $\mathbf{v} = (v_1, v_2)$ in the point \mathbf{x}_i ,
2. compute the tangent vector of abscissa given by the points \mathbf{x}_i and \mathbf{x}_F , i.e. $\mathbf{a} \leftarrow \mathbf{x}_i - \mathbf{x}_F$,
3. if $\mathbf{a} \cdot \mathbf{v}^\perp < 0$, change the direction of v^\perp , $v^\perp \leftarrow -v^\perp$
4. if $|\mathbf{v}^\perp| > \text{threshold}$, $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \frac{\mathbf{v}^\perp}{|\mathbf{v}^\perp|} h_d$, else $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \frac{\mathbf{a}}{|\mathbf{a}|} h_d$,
5. increase i by 1.

The initial condition built by the above algorithm allows us to segment much longer boundary curves in one step, see Fig. 2.1 right where yellow curve represents the initial condition and red curve the segmentation result.

2.2. The semi-implicit numerical scheme. Let us rewrite the intrinsic PDE (2.5) for the planar curve evolution into the following form

$$\mathbf{x}_t - \alpha \mathbf{x}_s = \delta \mathbf{x}_{ss} + w \mathbf{x}_s^\perp, \quad (2.8)$$

where $w = \lambda \mathbf{v}_N$ and α is given by (2.7). First, we perform the spatial discretization, which is based on the flowing finite volume method [12, 14] 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 [10, 11].

Integrating (2.8) over the finite volume $\mathbf{p}_i = [\mathbf{x}_{i-\frac{1}{2}}, \mathbf{x}_{i+\frac{1}{2}}]$, where $\mathbf{x}_{i-\frac{1}{2}}$ represents the middle point between the points \mathbf{x}_{i-1} and \mathbf{x}_i , i.e.

$$\mathbf{x}_{i-\frac{1}{2}} = \frac{\mathbf{x}_{i-1} + \mathbf{x}_i}{2}, \quad (2.9)$$

we get

$$\int_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \mathbf{x}_i ds - \alpha \int_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \mathbf{x}_s ds = \delta \int_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \mathbf{x}_{ss} ds + w \int_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \mathbf{x}_s^\perp ds, \quad (2.10)$$

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 (2.10), for $i = 2, \dots, n-1$,

$$\frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t - \alpha_i [\mathbf{x}]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} = \delta_i [\mathbf{x}_s]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} + w_i \left([\mathbf{x}]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} \right)^\perp \quad (2.11)$$

$$\frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t - \alpha_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right) = \delta_i [\mathbf{x}_s]_{\mathbf{x}_{i-\frac{1}{2}}}^{\mathbf{x}_{i+\frac{1}{2}}} + w_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right)^\perp. \quad (2.12)$$

Note, that \mathbf{x}_1 and \mathbf{x}_n are given by the Dirichlet boundary condition. 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,

$$\begin{aligned} & \frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t - \alpha_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right) \\ &= \delta_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h_{i+1}} - \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{h_i} \right) + w_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right)^\perp. \end{aligned} \quad (2.13)$$

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

$$-\alpha_i \left(\mathbf{x}_{i+\frac{1}{2}} - \mathbf{x}_{i-\frac{1}{2}} \right) = \alpha_i \left(\mathbf{x}_i - \mathbf{x}_{i+\frac{1}{2}} \right) - \alpha_i \left(\mathbf{x}_i - \mathbf{x}_{i-\frac{1}{2}} \right). \quad (2.14)$$

Using (2.9) we obtain

$$\begin{aligned} & \frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t + \frac{\alpha_i}{2} (\mathbf{x}_i - \mathbf{x}_{i+1}) - \frac{\alpha_i}{2} (\mathbf{x}_i - \mathbf{x}_{i-1}) \\ &= \delta_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h_{i+1}} - \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{h_i} \right) + w_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_{i-1}}{2} \right)^\perp. \end{aligned} \quad (2.15)$$

If $\alpha_i < 0$, i.e. the velocity of advection ($-\alpha$) in (2.8) 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

$$b_{i-\frac{1}{2}}^{in} = \max(-\alpha_i, 0), b_{i-\frac{1}{2}}^{out} = \min(-\alpha_i, 0), \quad (2.16)$$

$$b_{i+\frac{1}{2}}^{in} = \max(\alpha_i, 0), b_{i+\frac{1}{2}}^{out} = \min(\alpha_i, 0) \quad (2.17)$$

and rewrite the equation (2.15) as follows

$$\begin{aligned} & \frac{h_i + h_{i+1}}{2} (\mathbf{x}_i)_t + \frac{1}{2} \left(b_{i+\frac{1}{2}}^{in} + b_{i+\frac{1}{2}}^{out} \right) (\mathbf{x}_i - \mathbf{x}_{i+1}) + \frac{1}{2} \left(b_{i-\frac{1}{2}}^{in} + b_{i-\frac{1}{2}}^{out} \right) (\mathbf{x}_i - \mathbf{x}_{i-1}) \\ &= \delta_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{h_{i+1}} - \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{h_i} \right) + w_i \left(\frac{\mathbf{x}_{i+1} - \mathbf{x}_{i-1}}{2} \right)^\perp. \end{aligned} \quad (2.18)$$

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

$$\begin{aligned} & \frac{h_{i+1}^m + h_i^m}{2} \frac{\mathbf{x}_i^{m+1} - \mathbf{x}_i^m}{\tau} + \frac{1}{2} b_{i+\frac{1}{2}}^{in^m} (\mathbf{x}_i^{m+1} - \mathbf{x}_{i+1}^{m+1}) + \frac{1}{2} b_{i-\frac{1}{2}}^{in^m} (\mathbf{x}_i^{m+1} - \mathbf{x}_{i-1}^{m+1}) \\ & - \delta_i^m \left(\frac{\mathbf{x}_{i+1}^{m+1} - \mathbf{x}_i^{m+1}}{h_{i+1}^m} - \frac{\mathbf{x}_i^{m+1} - \mathbf{x}_{i-1}^{m+1}}{h_i^m} \right) \quad (2.19) \\ & = -\frac{1}{2} b_{i+\frac{1}{2}}^{out^m} (\mathbf{x}_i^m - \mathbf{x}_{i+1}^m) - \frac{1}{2} b_{i-\frac{1}{2}}^{out^m} (\mathbf{x}_i^m - \mathbf{x}_{i-1}^m) + w_i^m \left(\frac{\mathbf{x}_{i+1}^m - \mathbf{x}_{i-1}^m}{2} \right)^\perp, \end{aligned}$$

from where we get the fully discrete scheme in the form of a tridiagonal system

$$\begin{aligned} & \mathbf{x}_{i-1}^{m+1} \left(-\frac{\delta_i^m}{h_i^m} - \frac{b_{i-\frac{1}{2}}^{in^m}}{2} \right) + \mathbf{x}_{i+1}^{m+1} \left(-\frac{\delta_i^m}{h_{i+1}^m} - \frac{b_{i+\frac{1}{2}}^{in^m}}{2} \right) \\ & + \mathbf{x}_i^{m+1} \left(\frac{h_{i+1}^m + h_i^m}{2\tau} + \frac{\delta_i^m}{h_i^m} + \frac{\delta_i^m}{h_{i+1}^m} + \frac{b_{i-\frac{1}{2}}^{in^m}}{2} + \frac{b_{i+\frac{1}{2}}^{in^m}}{2} \right) = \mathbf{x}_i^m \frac{h_{i+1}^m + h_i^m}{2\tau} \quad (2.20) \\ & - \frac{b_{i+\frac{1}{2}}^{out^m}}{2} (\mathbf{x}_i^m - \mathbf{x}_{i+1}^m) - \frac{b_{i-\frac{1}{2}}^{out^m}}{2} (\mathbf{x}_i^m - \mathbf{x}_{i-1}^m) + w_i^m \left(\frac{\mathbf{x}_{i+1}^m - \mathbf{x}_{i-1}^m}{2} \right)^\perp, \end{aligned}$$

$i = 2, \dots, n-1$, where n is the number of the curve grid points. Let us note the the points \mathbf{x}_1 and \mathbf{x}_n are prescribed by the Dirichlet boundary conditions.

The system (2.20) is represented by a strictly diagonally dominant matrix, thus it is always solvable by the classical Thomas algorithm and the solvability is guaranteed without any restriction on the time step length τ [15].

In the numerical scheme (2.20) there are two parameters w_i^m and α_i^m . Parameter w_i^m is evaluated as follows,

$$w_i^m = \lambda_i^m v_{\mathbf{N}_i^m}, \quad (2.21)$$

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 $v_{\mathbf{N}_i^m}$ is the discrete value of the function $v_{\mathbf{N}}$ in the curve point \mathbf{x}_i^m .

In order to discretize the tangential velocity α we first set $\alpha_1^m = 0$, which ensures the point \mathbf{x}_1 will not move in the tangential direction. Then, by the finite volume discretization of equation (2.7) we get α_i^m for $i = 2, 3, \dots, n$ by

$$\alpha_i^m = \alpha_{i-1}^m + h_i^m k_i^m \beta_i^m - h_i^m \langle k\beta \rangle_\Gamma + \omega \left(\frac{L^m}{n} - h_i^m \right), \quad (2.22)$$

where the curvature k_i^m , the normal velocity β_i^m , for $i = 2, 3, \dots, n$, the mean value $\langle k\beta \rangle_\Gamma$ and the total curve length L^m are given by following formulas

$$k_i^m = \text{sgn}(\mathbf{h}_{i-1}^m \wedge \mathbf{h}_{i+1}^m) \frac{1}{2h_i^m} \arccos \left(\frac{\mathbf{h}_{i+1}^m \cdot \mathbf{h}_{i-1}^m}{h_{i+1}^m h_{i-1}^m} \right), \quad (2.23)$$

$$\beta_i^m = \frac{\delta_{i-1}^m + \delta_i^m}{2} k_i^m + \frac{w_{i-1}^m + w_i^m}{2}, \quad (2.24)$$

$$\langle k\beta \rangle_\Gamma = \frac{1}{L^m} \sum_{l=2}^n h_l^m k_l^m \beta_l^m, \quad L^m = \sum_{l=2}^n h_l^m, \quad (2.25)$$

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

2.3. The final adjustment of the segmentation. Usually, the results of semi-automatic segmentation need some final adjustment. The user may slightly misclick the end point of the segment which becomes the first point of the next segment. This misclick may lead to a non-smooth segments interconnection, see yellow curve on Fig. 2.2 left. Similarly, using a very low curvature influence during the segmentation, the result can be "pixelized", see again yellow curve on Fig. 2.2 right. In both cases we need a smoothing of the curve but together with its precise adjustment to the edges.



FIG. 2.2. *The final adjustments of the semi-automatic segmentation curve. Left: when the segments are not smoothly interconnected, right: when the segmentation result is "pixelized".*

To perform the final adjustments we consider the governing equation (2.5) applied to the whole curve consisted of interconnected segments as the initial condition, either with Dirichlet boundary conditions (if the interconnected curve is still open) or with the periodic boundary conditions (if the curve is closed). We use again the semi-implicit scheme described in the previous section, solving the tridiagonal system by the classical Thomas algorithm (in case of open curve) or by its modification to the cyclic tridiagonal solver (in case of closed curve), and just one time step of the scheme is usually sufficient to get accurate and smooth final adjusted result, see red curves in Fig 2.2.

3. NaturaSat software overview. The semi-automatic segmentation is an integral and one of the most important parts of the NaturaSat software for exploration of Natura 2000 habitats. This software has been built in cooperation with ESA - European Space Agency and Institute of Botany of the Plant Science and Biodiversity Center, Slovak Academy of Sciences and supported mainly by the ESA-PECS contract No. 4000122575/17/NL/SC and project APVV-16-0431. NaturaSat is built in C++ language using various open-source libraries. The graphical user interface (GUI) is created using Qt5 framework [18] to allow uniform codebase and look on different platforms. To read Sentinel-2 datasets we use GDAL library [5] and Zlib library [4], for the conversion between different coordinate systems we use GeographicLib [8] and for the numerical linear algebra solvers we use Eigen3 template library [6].

The current GUI, see Fig. 3.1, is divided into four main sections: Workspace toolbar, Tools toolbar, Preview map viewer and Data viewer. The main purpose of the Data viewer widget is to render loaded data. We are able to zoom and move through data using mouse commands performed over the widget area. The current zoom level is displayed under the widget area together with the current geographic coordinates (in WGS84 and UTM systems) of a point under the mouse cursor. In the widget, there is visible only a part of the entire area of the loaded data (according to the zoom level), therefore, for a better orientation in the loaded data a user can

use Preview map viewer widget. In this widget, there are two highlighted rectangular areas. The light blue rectangle represents the area which is currently displayed in the Data viewer and the orange rectangle represents the so-called working area. Since the loaded data has very high resolution, performing segmentation on the entire area would be highly memory consuming, therefore all implemented segmentation tools are applied only on the working area which is still large enough to perform segmentation of extensive structures.

The NaturaSat software is able to load multiple Sentinel-2 datasets in zip files either downloaded from Copernicus Open Access Hub [2] outside the software or using the NaturaSat remote data manager. To access Sentinel-2 data repositories directly from NaturaSat we use OpenSearch API on Copernicus Open Access Hub [3]. Content of the API request query link depends on user input in the user application interface. Since the NaturaSat project tasks are currently performed on data from the area of Slovakia, we divided this area according to predefined Sentinel-2 tiling grid. The tiling grid contains approximately 56 500 tiles worldwide with defined code names and 11 of them belong to the area of Slovakia. We use the code names for the identification of the user-selected area. Each tile represents an area of $100 \times 100 \text{ km}^2$ in ortho-images in UTM/WGS84 projection. In the application interface the user can select one of the available tiles covering Slovakia and two dates defining a scope of data sensing dates. Since our segmentation algorithms are designed for optical data of sufficient quality the user is able to select also a maximal cloud cover percentage.

The list of all loaded Sentinel-2 datasets is displayed in the top area of the Workspace toolbar. Here the user can easily switch between all loaded datasets or unload them from the application. The satellite optical bands (e.g. B2 - Blue, B3 - Green, B4 - Red, B5 - Vegetation classification, B8 - Near-infrared etc.) can be freely switched to any combination without constrains regarding the band resolution. For a better visualization of loaded data, we need to slightly modify the histogram of image intensities. The software contains a crop histogram tool where the user can set a percentage value of the crop parameter and remove the corresponding number of very low or very high intensities from the image. The software distribution includes a database of specific locations provided by botany specialists. These locations indicate areas where one or more important Natura 2000 habitats are located in Slovakia. In Data viewer, all of these locations (habitat markers) inside the loaded area are visualized using a color selected by the user in the Workspace toolbar. The user can also filter the habitat markers to show only locations of particular habitat type.

In addition to data manipulations, Workspace toolbar allows to load external curves or save and manipulate the created ones. The software can currently handle curves written in multiple data formats (e.g. *.kml, *.gpx, *.vtk, etc.). When a curve is created or loaded, a user can modify the name and color of this curve and can change its visibility state. A list of curves is visible in Workspace toolbar and is attached to the loaded data, therefore the application allows to create multiple curve lists, one for each loaded dataset.

The Tools toolbar contains four fundamental software tools: Filtering tool, Semi-automatic segmentation tool, Automatic segmentation tool and Monitoring tool.

The Filtering tool provides filtering functionality. We implemented four filtering methods widely used in image processing, namely filtering based on diffusion equations (linear diffusion and regularized Perona-Malik nonlinear diffusion) and the curvature-driven filtering methods (mean curvature flow and geodesic mean curvature flow models). All important parameters of each of the filtering methods can be set by

the user. This tool allows a user to see a filtered image beside original loaded data in the Data viewer. This provides the possibility to compare the effect of chosen filtering method and to tune parameters of the filtering with respect to the segmentation result.

The Semi-automatic segmentation tool contains algorithms described in the previous section of this paper. The user clicks and drags mouse cursor along the edge of the habitat boundary and the curve is automatically attracted to the edge. Each of the newly constructed segmentation curves (can be closed or open) is appended to the curve list in Workspace toolbar and can be further adjusted according to section 2.3. During construction of the curve, the software allows to remove the last segment and continue from the previous one or to continue with a previously unfinished open curve.

The Automatic segmentation tool offers the fully automatic segmentation method [17]. This method is based on the evolution of the initial curve toward the boundary of the segmented area. The software supports the construction of the initial curve as a circle with centre defined by clicking into Data viewer widget or as the circle with center in a selected habitat marker. In contrast to the semi-automatic segmentation, the automatic segmentation method depends on more input parameters. However, each of the important model inputs, such as the homogeneity detector, the expanding force and the velocity vector field (see [17]) can be displayed beside the loaded data in Data viewer widget for a better experience with tuning the model parameters.

The last of the currently implemented software tools is the Monitoring tool. The application provides the capability to compute several statistical characteristics (mean, standard deviation, minimum, maximum) of the optical data inside the area encompassed by a selected curve through the Statistical interface. Further (differential) parameters such as the area, perimeter, isoperimetric ratio or Relative High Laplacian can be computed as well. The software can compute all these characteristics for selected bands and selected curves and display them in the structured table. This table can be exported to comma separate values (*.csv) format file or copied to Excel datasheet. The main idea of monitoring is based on the comparison of statistical characteristics for the same curve but for the data from different sensing periods. Such comparison is used for a habitat quality state monitoring in time. For a difference in the habitat area in time, the Statistical interface computes the Hausdorff distances between two boundary curves obtained in different times. All these capabilities of the Monitoring tool brings a possibility to follow Natura 2000 habitat changes in time and to inform responsible organizations in case of emergency.

Acknowledgments. We are grateful to colleagues from the Institute of Botany of the Plant Science and Biodiversity Center, Slovak Academy of Sciences for many fruitful discussions on the NaturaSat software design and its application to Natura 2000 habitat segmentation and monitoring.

REFERENCES

- [1] M.Ambroz, M.Balažovjeh, M.Medla, K.Mikula, *Numerical modeling of wildland surface fire propagation by evolving surface curves*, Advances in Computational Mathematics, April 2019, Volume 45, Issue 2, pp. 1067-1103
- [2] European Space Agency, *Copernicus Open Access Hub*, (2020). URL <https://scihub.copernicus.eu/>
- [3] European Space Agency, *OpenSearch API - Copernicus Open Access Hub*, (2020). URL <https://scihub.copernicus.eu/userguide/OpenSearchAPI>



FIG. 3.1. Graphical user interface of NaturaSat application

- [4] Gailly, J., Adler, M, *Zlib compression library*, (2004). URL <https://zlib.net/>
- [5] GDAL/OGR contributors, *GDAL/OGR Geospatial Data Abstraction software Library* Open Source Geospatial Foundation, (2020). URL <https://gdal.org>
- [6] G. Guennebaud, B. Jacob, *Eigen v3*, (2020). URL <http://eigen.tuxfamily.org>
- [7] T.Y.Hou, J.Lowengrub, M.Shelley, *Removing the stiffness from interfacial flows and surface tension*, J. Comput. Phys., 114 (1994) pp. 312-338.
- [8] Ch. F. F. Karney, *GeographicLib*, (2019), URL <https://geographiclib.sourceforge.io/>
- [9] M.Kimura, *Numerical analysis for moving boundary problems using the boundary tracking method*, Japan J. Indust. Appl. Math., 14 (1997), pp. 373-398.
- [10] K.Mikula, M.Ohlberger, *Inflow-Implicit/Outflow-Explicit Scheme for Solving Advection Equations*, in Finite Volumes in Complex Applications VI, Problems and Perspectives, Eds. J.Fort et al. (Proceedings of the Sixth International Conference on Finite Volumes in Complex Applications, Prague, June 6-10, 2011), Springer Verlag, 2011, pp. 683-692
- [11] K.Mikula, M.Ohlberger, J.Urbán, *Inflow-Implicit/Outflow-Explicit Finite Volume Methods for Solving Advection Equations*, Applied Numerical Mathematics, Vol. 85 (2014) pp. 16-37
- [12] K.Mikula, D.Ševčovič, *Evolution of plane curves driven by a nonlinear function of curvature and anisotropy*, SIAM J. Appl. Math., 61 (2001), pp. 1473–1501.
- [13] K.Mikula, D.Ševčovič, *A direct method for solving an anisotropic mean curvature flow of planar curve with an external force*, Mathematical Methods in Applied Sciences, Vol. 27, No. 13 (2004) pp. 1545-1565
- [14] K.Mikula, D.Ševčovič, M.Balažovjech, *A simple, fast and stabilized flowing finite volume method for solving general curve evolution equations*, Communications in Computational Physics, Vol. 7, No. 1 (2010) pp. 195-211
- [15] K.Mikula, J.Urbán, *New fast and stable Lagrangean method for image segmentation*, Proceedings of the 5th International congress on image and signal processing (CISP 2012), Chongqing, China, October 16-18, 2012, ISBN-978-1-4673-9/10, pp. 834-842
- [16] K.Mikula, J.Urbán, M.Kollár, M.Ambroz, I.Jarolimek, J.Šibik, M.Šibiková, *Semi-automatic segmentation of NATURA 2000 habitats in Sentinel-2 satellite images by evolving open curves*, Discrete and Continuous Dynamical Systems Series S, accepted
- [17] K.Mikula, J.Urbán, M.Kollár, M.Ambroz, I.Jarolimek, J.Šibik, M.Šibiková, *An automated segmentation of NATURA 2000 habitats from Sentinel-2 optical data*, Discrete and Continuous Dynamical Systems Series S, accepted
- [18] The Qt Company, *Qt: Cross-platform software development for embedded & desktop*, (2020). URL <https://www.qt.io/>