

GRZEGORZ LENDA, PIOTR BANASIK
AGH University of Science and Technology
Faculty of Geo-Data Science Geodesy and Environmental Engineering
Kraków, Poland
orcid.org/0000-0001-8876-1055; grzenda@agh.edu.pl
orcid.org/0000-0002-3604-4019; pbanasik@agh.edu.pl

A new “walking spline” method for smoothing polylines on the maps using B-spline functions based on an iteratively optimised set of vertices

Abstract. This research paper proposes a new method for smoothing polylines on a map, using B-splines iteratively moving away from the vertices. Splines are a popular tool used to smooth data. Unfortunately, natural objects represented by polylines, usually have an irregular shape and unevenly spaced vertices. For such data, splines may have a distorted course. The presented method changes an approach to polylines on the map. The smoothed line is no longer just a line connecting a set of vertices on the map but becomes a path that a hypothetical hiker must follow. The hiker attempts to optimise (simplify) the route so that, it requires the least amount of effort. The hiker will therefore avoid the vertices at a certain distance, leaving dense and even traces. These traces constitute a data set for which the unfavourable properties of splines are eliminated. The vertices of the polyline will be smoothed (bypassed) by the B-spline control point function, which, spread over a dense set of tracks, will perform the first, slight smoothing. Subsequent smoothing iterations will be performed on a modified, increasingly sparse, but still uniform set of traces from the previous iterations. This will allow for even smoothing of any polyline.

Keywords: mapping, map, cartographic generalisation, smoothing, B-spline

1. Introduction

The study presents a method for smoothing polylines on the map, based on B-spline functions, determined for appropriately modified input data. Spline functions accurately represent shapes with continuous curvature, therefore they are potentially the right tool for smoothing lines and surfaces. Unfortunately, natural objects represented by polylines on the map, usually have an irregular shape and unevenly spaced vertices. For such data, spline functions may have a distorted course. However, changing the approach towards the set of data to be smoothed: from the set of vertices that need to be connected to the set of steps that need to be taken to complete a given route, any

complex input polyline can be smoothed using iterative B-spline functions. Smoothing is continued until the specified tolerance for the distance between the new curve and the points of the input curve is exceeded. The value of such a distance may result, for example, from the target scale of the map (Kuna et al., 2024). Each subsequent iteration results in better smoothing, and if the tolerance is not specified, it continues until a straight line is created between the starting and ending points of the line. The method is simple to use, effective in smoothing and does not require manipulation of various parameters. To understand its idea, it is necessary to discuss two issues in detail. The first one is related to the sources of errors arising when describing irregular objects using

spline functions and presenting the optimal case, in which these errors are minimised. The second issue concerns the idea of replacing the few vertices of the polyline with more densely spaced points. These points may symbolise the steps of a hypothetical hiker following the route defined by the polyline of the trail. A hiker who wants to complete the trail with the least effort, will try to shorten it wherever possible, but so, as not to lose the trail. This route optimisation produces a data set, for which, the spline function is created in optimal conditions that eliminate distortions.

The innovation of the method should be considered from the perspective of the effectiveness of smoothing lines on a map. The method combines appropriately selected spline functions with the observation of an intuitive tendency to smooth out the route while also shortening it. Line smoothing on maps is a special type of smoothing that should take into account, among other things, the cartometric feature of the map. When reducing the scale of a map, smoothing should simplify the shape of the polyline while maintaining its characteristic shape. Smoothing lines on maps should move the curve relatively uniformly away from the polyline, ensuring the smoothest possible course without any undulations. Standard methods for creating splines from chaotic data sets do not allow this, as demonstrated in the theoretical section. The spline functions used to smooth lines on maps (Steiniger & Meier, 2004; Li, 2007a; Guo et al., 2017) encounter difficulties in selecting the appropriate parameters, which can lead to curve undulations. The proposed method avoids these difficulties and is simple to implement. This simplicity was achieved through an observation that allowed for the combination of two factors: the analysis of various properties of spline functions and the replacement of the smoothed line with an iteratively traversed route, using small and uniform steps. Without this combination, the demonstrated effectiveness of smoothing, as seen in the examples, would not have been possible. A comparison of shapes showed that the proposed method, as the degree of smoothing increases, outperforms the results obtained using the Polynomial Approximation with Exponential Kernel (PAEK) method, which is the standard tool used for smoothing lines on a map. The advantage of the proposed method lies in

having a single parameter that can naturally be linked to the recognizability of length on a map at a given scale.

Below, after reviewing the literature related to the generalisation of the polylines, which includes, inter alia, simplification and smoothing, the details of the method have been discussed, and examples of its effects have been presented. Finally, it has been compared with the popular smoothing method PAEK (Bodansky et al., 2002), used e.g. in the ARC GIS package (Esri, 2024).

2. Literature review

The generalisation of information provided in the form of a cartographic image (map), occurs at every stage of its creation, from the stage of capturing information from the real space until the stage of final presentation (Cebrykow, 2017). A map is a conventional model of reality, and generalisation is one of the foundations of this model. It results from the need to use a scale $m < 1$, so that a much larger part of the actual Earth's surface can be represented on its small area. Two directions have been distinguished in digital cartographic generalisation. The first one concerns the generalisation of the image (map), and the second one concerns the generalisation of spatial data contained in digital databases (Weibel & Jones, 1998). Contemporary generalisation seems to be heading in the latter direction (Muller, 1991).

Cartographic generalisation involves reducing the amount and detail of the information provided. This is achieved by aggregating, simplifying (generalising) information, but also by strengthening and magnifying it (McMaster & Shea, 1992). One of the first attempts to generalise map content concerned objects represented using lines (rivers, roads, isolines, borders, etc.). It involved simplifying the shape of the line, mainly by reducing its vertices and smoothing its course. The shape should be simplified, while maintaining the accuracy of the position and line alignment appropriate to the scale of the map. This results from the cartometric feature of the map and is especially important in the case of large-scale maps. It seems that generalisation of lines should be carried out through the combined use of the simplification operator and additional shape smoothing (Bourghardt, 2005).

Over the past 70 years, many concepts and algorithms for curve simplification have been developed (Heckbert & Garland, 1997; Luebke, 2001; Li, 2007a; Weibel, 2020; Baig et al., 2013). Some of them have become a tool in GIS and CAD software, to extend the zoom service on web maps (Neun et al., 2009). Such a service requires, among other things, automatic simplification and smoothing of curves. This can be achieved by using objective criteria and unambiguous, simple parameters in simplification algorithms (Bac-Bronowicz et al., 2023).

Smoothing is treated as one of the shape generalisation operators (Foerster et al., 2007). It allows for the removal of visible irregularities and line breaks, which improves the aesthetics and related readability of the map. Smoothing can be applied to curves that have been subjected to initial simplification or as a stand-alone method (Tutić & Lapaine, 2010). Various methods are used for smoothing, using e.g. the fast Fourier transform (Li, 2007a; Shan et al., 2023), wavelets (Li, 2007b; Saux, 2003), approximation polynomials (Li, 2007b; Spoerhase, 2019) including the PAEK method (Bodansky et al., 2002; Esri, 2024) or Bezier curves (Nöllenburg, 2008), empirical mode decomposition EMD (Li, 2007b; Li et al., 2017), DEM smoothing (Zaksek & Podobnikar, 2005; Nieuwenhuizen et al., 2021; Kettunen et al., 2017). Some of these methods work in an iterative mode, searching for optimal smoothing parameters. Comprehensive testing of methods should take into account many features of algorithms (Kuna et al., 2024).

Spline functions play an important role in smoothing lines. They implement a polynomial description of a line divided into fragments. Spline functions applied to a polyline can simplify its course and ensure its smooth change. Both properties are implemented in accordance with the principle of energy minimisation (de Boor, 1978). The first attempts to use splines were concerned with smoothing the edges of objects placed in images (Kass et al., 1988). This work only examined the possibilities of splines in graphic transformations. An example of a cartographic application was the use of splines to smooth polylines symbolising polylines on a map (e.g. rivers, roads) (Bourghardt, 2005). Attention was paid to the problem of automatic selection of function parameters and to the problem of moving endpoints of a line.

Multi-stage smoothing has also been proposed. Subsequent attempts to use splines concerned smoothing the set of lines in the form of isobaths on nautical maps (Guilbert & Lin, 2006; Guilbert & Saux, 2008). Smoothing for a set of lines had an additional requirement to maintain the co-shape of adjacent lines. In these works, the undesirable effect of generalisation, consisting in collisions of simplified lines with neighbouring lines, was also examined, and an appropriate method for correcting this phenomenon was proposed. Snake spline curves (Steiniger & Meier, 2004) have great smoothing possibilities, thanks to the use of parameters related to the internal and external energy of the spline, determining its stress. Unfortunately, determining the optimal values of these parameters is difficult and time-consuming in practice (Li, 2007a; Guo et al., 2017; Velut et al., 2007; Brigger et al., 2000). Another approach to smoothing isolines with splines is presented in (Bayer et al., 2023). Splines are used to smooth dense point sets (ALS point clouds) from which polylines were generated (Jiang et al. 2023; Bodansky et al., 2002). The advantage of this type of data is the ability to define lines using multiple points. The large number of regularly spaced points that make up a line, improves the effectiveness of splines in smoothing its shape. Such a definition of a line to be smoothed on maps is not common. Most topographic databases used to create maps contain limited point data. In these databases, lines symbolising rivers, roads and borders are defined only by those points that are necessary to represent shape variability.

However, spline functions exhibit certain properties that make smoothing irregular shapes difficult. The influence of these properties is discussed below, and the *walking spline* method is proposed, which allows the smoothed model to be independent of them.

3. Problems associated with approximating irregular data by spline functions

One of the tools used to smooth data are spline functions. However, it is not a tool designed for smoothing polylines and therefore its effects are not always desirable. Spline functions were created as the mathematical equivalent of the flexible strips used to design shapes with continuous curvature in the pre-

computer era. They are determined based on the Bernoulli-Euler equation describing the curvature of a beam, bending in the elastic range (Ahlberg et al., 1967; de Boor, 1978).

$$K = \frac{f''(x)}{[1+(f'(x))^2]^{\frac{3}{2}}} = \frac{M_z}{EJ} \quad (1)$$

where: K – curvature, M_z – bending moment, E – Young’s modulus, J – moment of inertia

Because deflections of a beam bending in the elastic range are small, it is assumed that the first derivative is zero, resulting a simplified equation:

$$f''(x) = \frac{M_z}{EJ} \quad (2)$$

from which the relations enabling the determination of the spline function are further derived.

Lines on a map (e.g. roads, rivers, borders), often appear as polylines with complicated course, and do not have continuous curvature. At the stage of obtaining the points forming the line, only characteristic points of changing its course are selected. These points are connected by segments, creating a polyline. It cannot be assumed that “deflections”, i.e. changes in the inclination of subsequent sections, will be small, either. In such cases, the spline function undergoes deformations (undulations) (Farin, 2002; Floater & Surazhsky, 2006; Piegł & Tiller, 2012), illustrated in Figure 1, where local extrema of the S function appear next to some points (p_2 and p_3).

Another problem with the accuracy of spline modelling is related to data density. In (Birkhoff

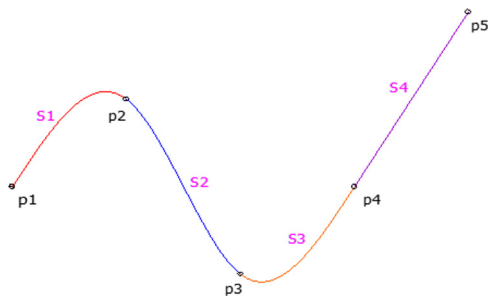


Figure 1. Undulations of a spline function interpolating a set of highly variable points (local extrema of the S function appear next to points p_2 and p_3)

& de Boor, 1964) the convergence of the spline function S to the approximate function f was estimated. For commonly used spline functions of class C^2 , this convergence will depend quadratic on the distance between the points:

$$|f - S| \leq \max(h_i^2)M \quad (3)$$

where: $M = \text{const}$, $M \geq |f''(x)|$, $h_i = x_i - x_{i-1}$

Therefore, the increase in data density significantly affects the accuracy of the approximations. The spline function drawn through 6 points (Figure 2b) provides a much better approximation of the circle than the function drawn through 3 points (Figure 2a).

A polyline on the map consists of the minimum number of vertices necessary to correctly represent its shape, which contributes to the occurrence of distortions visible in Figure 2a. Additionally, these lines may have different vertex densities in different places. This is another problem when modelling with spline functions. Each spline function is created based on knots. Appropriate selection of knots (parametrisation) is essential for the accuracy of the approximation if the set of points is irregular. The most common are three knot selection methods (Piegł & Tiller, 2012; Haron et al., 2012; Kiciak, 2019; Peterson, 2006): equidistant (4), proportional to the distance $d_{i,i-1}$ between points (5) and proportional to the square root distances between points.

$$T_{1i} = \text{const} \quad (4)$$

$$T_{2i} = d_{i,i-1} \quad (5)$$

$$T_{3i} = \sqrt{d_{i,i-1}} \quad (6)$$

where: T – node selection method, d – length of the section between neighbouring nodes $i, i - 1$

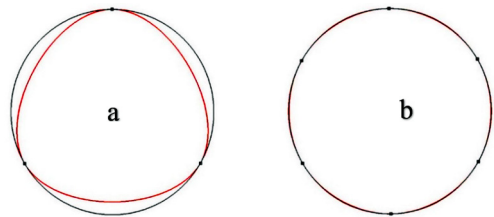


Figure 2. Spline function approximating a circle: a) based on 3 points; b) based on 6 points

For a non-uniform set of points, the simplest parametrisation assuming equidistant knots brings the worst results. Distance parametrisation produces the best results when approximating shapes with slowly changing curvature, and root-distance parametrisation yields the best results when approximating shapes with quickly changing curvature. If the points are equidistant, all parametrisations give the same

the beam on adjacent sections are smaller and smaller relative to each other. This is increasingly consistent with the theoretical assumptions of spline functions. As a result, the spline function better approximates the determined shape b.1) – b.3).

From the point of view of smoothing the polyline on a map, the use of B-spline functions can be considered (Farin, 2002; Piegl & Tiller,

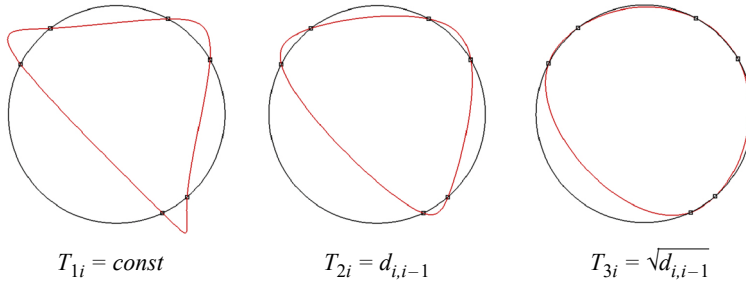


Figure 3. Approximation of a circle by a spline function based on non-uniformly distributed points: parametrised by a constant distance, the square root of the distance between points and the distance between points

result. Figure 3 illustrates spline functions approximating a circle based on six non-uniformly selected points, parametrised by the above-mentioned methods. It demonstrates how important the choice of parametrisation is and that even the best one cannot fully compensate the distortions resulting from the lack of regularity of the points.

The analysis of Figures 2 and 3 reveals, that in order to provide a good approximation of the modelled shape of the line, it is necessary to provide a relatively large set of evenly spaced points. The set of vertices of a polyline on a map usually does not meet these conditions.

The importance of a dense set of points for the accuracy of approximations is further explained in Figure 4. It is noticeable that for points sparsely distributed on the circle a.1), the angle between the three points is acute. The question arises whether in such a case the theoretical conditions of beam deflection mechanics in terms of elasticity could still be met. This is all the more important because, when deriving formulas for determining the spline function, small deflections are assumed, reducing equation (1) to form (2). However, as the density of points increases, the angles between them increase (a.2 and a.3), i.e. the “deflections” of

2012; Schumaker, 2007), because they offer the possibility of creating curves in two ways: based on interpolation points or control points (B-spline interpolated curve and B-spline control point curve). The B-spline function described by equation (7) is a linear combination of base functions N , defined by the Mansfield-de Boor-Cox formula, of m degree, whose coefficients are control points d , creating a control polyline (Figure 5).

$$S_i(t) = \sum_{i=0}^{n-m-1} d_i N_i^m(t) \quad t_i = \{t_0, \dots, t_n\} - \text{knots} \quad (7)$$

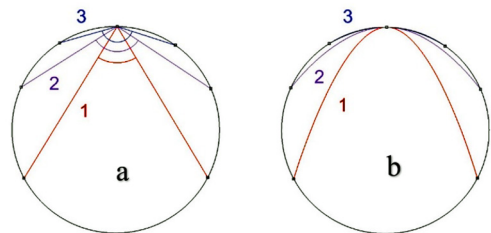


Figure 4. An increase in the density of points (1,2,3) results in an increase in the angles between adjacent sections (a), thanks to which the model is locally less and less deflected, which improves the approximation of the determined shape (b).

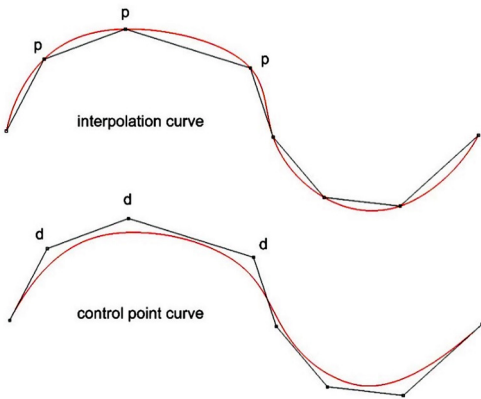


Figure 5. Difference between a B-spline curve based on interpolation points (p) and control points (d). In the figures, the vertices of the same polyline are taken as points p for creating an interpolation curve or as d for creating a curve from control points.

The B-spline function created based on interpolation points p passes through these points and the control points d are determined unknowns. The interpolation curve is located outside the convex parts of the polyline connecting the points p . The B-spline function created from the control points d has the vertices of the control polyline at these points, and the B-spline is inside the convex parts of the control polyline (Figure 5).

The curve from the interpolation points passes directly through the points, just like all the curves presented in the previous figures.

Figure 6 illustrates a curve with interpolation points against a polyline, which may be an irregular polyline on a map, that can be smoothed. The polyline has been smoothed, but the spline function has numerous undulations. When smoothing a polyline on a map, its simplification is expected at the same time. However, the interpolation spline function complicates it in certain places. A smoothed curve will always be longer than a polyline because straight lines between points are replaced by arcs that are based on the same points. The B-spline function based on control points, allows for smoothing combined with simplification of the shape (Figure 7). This curve has several important features as far as smoothing of polylines on map is concerned. The curve is always inside the convex outline, i.e. it does not go beyond the outer outline of the convex parts. Thus, it is always shorter than the smoothed polyline. Unlike a curve with interpolation points, it simplifies the input shape. All vertex points of the polyline are smoothed independently of each other, and the end points are preserved. Unfortunately, smoothing is performed in an unintuitive and uneven way, which results from irregular lengths of sections and vertical angles.

In general, B-spline functions created based on control points have features useful from the point of view of polyline smoothing, but they are also characterised by significant and disproportionate smoothing of some vertices. Therefore, they cannot be used directly for

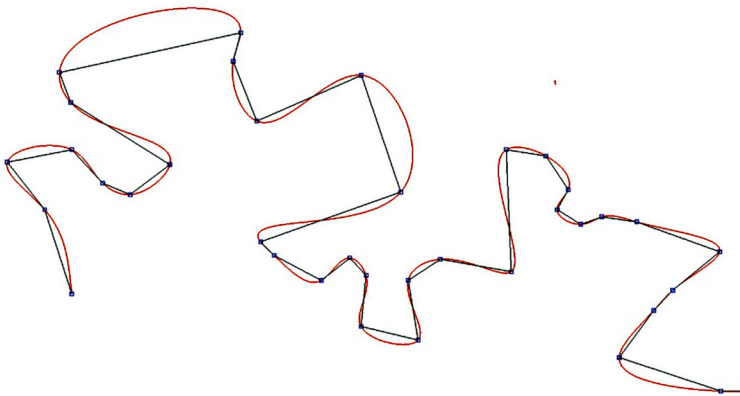


Figure 6. An interpolating B-spline function presented against a smoothed irregular polyline. The spline is 6% longer than the broken line

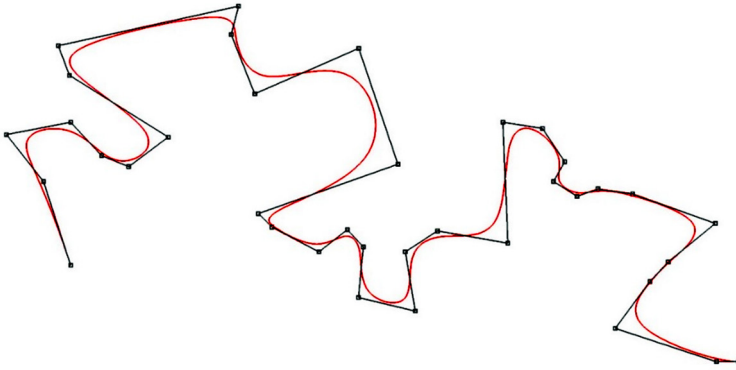


Figure 7. A B-spline function from control points presented against a smoothed irregular polyline.
The spline is 19% shorter than the broken line

smoothing polylines, especially when the smoothing is to have small values. Additionally, a B-spline curve from control points based on the vertices of the smoothed polyline gives only a single smoothing, which would be difficult to adjust, e.g. to the selected map scale.

When analysing the presented properties of spline functions, it is possible to distinguish certain conditions that would allow for even smoothing of polylines on a map. Smoothing can be performed using a B-spline function from control points, but under conditions that ensure a relatively slow and even divergence from the vertices. To meet these conditions, the points forming the line should be arranged as densely and regularly as possible.

4. Changing approach to the smoothed polyline on a map

The second issue related to the proposed smoothing method concerns changing the approach to the set of input data, that constitute the input polyline, that is to be smoothed. Classically, when looking at a map, for each polyline we see a set of vertices that are located in characteristic places of a given object. This is where the current direction of the line changes. Depending on the shape of the object, the vertices may be located with different density, and the lines in between can create any angles. From the considerations devoted to spline functions, it is known, that these are situations

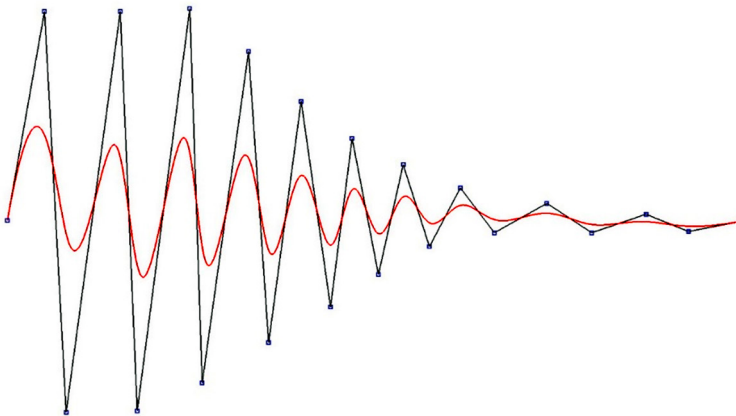


Figure 8. A B-spline function created from control points presented against a smoothed polyline. The fastest divergence from the vertices occurs between the longest sections creating the most acute angles

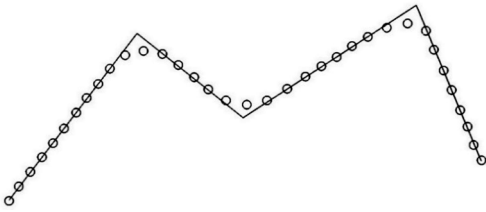


Figure 9. A polyline of the trail traversed by the hiker using dense and equidistant steps (circles)

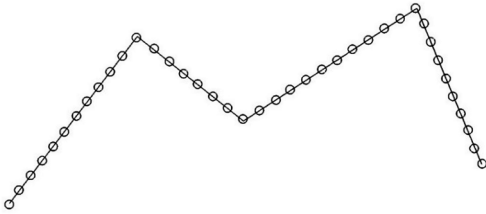


Figure 10. A shortened trail performed by a hiker, which involves bypassing vertices (landmarks) at a certain distance on the inside of the trail

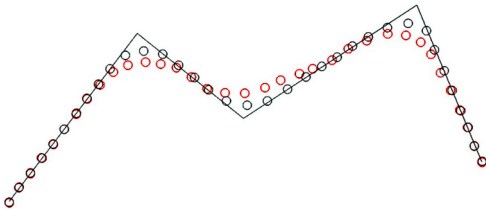


Figure 11. Red traces left by the next hiker, who shortened the trail marked by his predecessor, bypassing the black traces on the inside of the trail vertices

favouring the occurrence of undulations in spline functions.

However, the set of vertices of a polyline can be viewed from another perspective: for example, a tourist traversing a marked trail along such a polyline. A polyline presented on a map symbolises a certain route. However, when the same route is to be walked, with some effort, its optimisation, so that the least effort is required, will be natural. Polyline vertices are points in the field, where the change in the direction of the route is marked. Hiker traversing the trail reach these points, then change direction. The hiker moves in steps, leaving behind traces that are equidistant and much more densely spaced than the marked points on the trail (Figure 9).

Converting vertices into steps does not change the shape of the route but only increases the density and regularity of the input data. From the previous considerations, it is known that such a data distribution eliminates the unfavourable properties of spline functions. However, the hiker does not need to go directly to the vertices but can bypass them at a certain distance. Wanting to shorten (optimise) the total walking route, he will lead the path on the inside of the vertices (Figure 10).

In practice, this smooths and simplifies the polyline of the trail. This is the first smoothing pass, keeping the curve close to the initial vertices. It can be said that the hiker moves relative to the trail landmarks in the same way as a B-spline function moves relative to control points, with the difference, that the hiker approaches all vertices in a similar, predictable way. The traces left by the hiker will become new landmarks of the route. The next tourist, in order to shorten the route marked by his predecessor, will follow his steps, bypassing them at a certain distance on the inside of the previous vertices (Figure 11).

The shortcuts made by the second hiker will therefore be the shortcut of the first hiker's shortcut. The next hiker will follow this path, introducing his own shortcut to the previously created shortcuts. In this way, subsequent paths will move away from the initial set of trail landmarks in a similar way. After many passes, the path will turn into a straight line connecting the beginning and end of the trail. However, subsequent paths, based on an increasingly smoother route, will move away from the initial vertices very slowly. Therefore, shortening the route will be ineffective and many subsequent crossings will be necessary to simplify the route as much as possible. To speed up this process, let us assume that each subsequent hiker will be taller than his predecessor, with increasingly larger steps. Each new path will therefore move further and further away from the landmarks of its predecessor.

5. Smoothing polylines on maps using B-spline functions created on hiker's traces

The principle described above, natural from the point of view of a hiker who wants to shorten (optimise) the trail, can be combined with spline

functions that will smooth the route near the bypassed vertices. Since the splines will be created on dense and regular traces, they will not be sensitive to the distortions described earlier. The B-spline function, created based on a control polyline, has the feature of bypassing vertices from the inside, but it is done not very intuitively and is difficult to control in practice. In Figures 7 and 8, it could be seen that depending on the distribution of vertices, the B-spline function can diverge from them quickly or slowly, causing disproportionate smoothing of the control polyline. This depends on the differences in the lengths of adjacent sections and the angles between them. However, replacing the vertices of the polyline with steps solves this problem. The segments have the same lengths, so there are no differences in the lengths of adjacent segments. Acute angles between adjacent segments will

only occur near the original polyline vertices and will be included between short segments. Therefore, the B-spline function will not be able to significantly smooth some sections in the first iteration. In subsequent iterations, the set of input points will be obtained from a previously created curve that has already been smoothed. The angles between successive sections will therefore become more and more similar to each other, reducing the effect of disproportionate smoothing of the polyline by the B-spline function.

Replacing vertices with dense and uniform steps therefore allows the creation of B-spline functions from control points under optimal conditions. Subsequent small increment iterations prevent the B-spline function from suddenly and uncontrollably smoothing some parts of the line. The method presented during theoretical considerations works as detailed below:

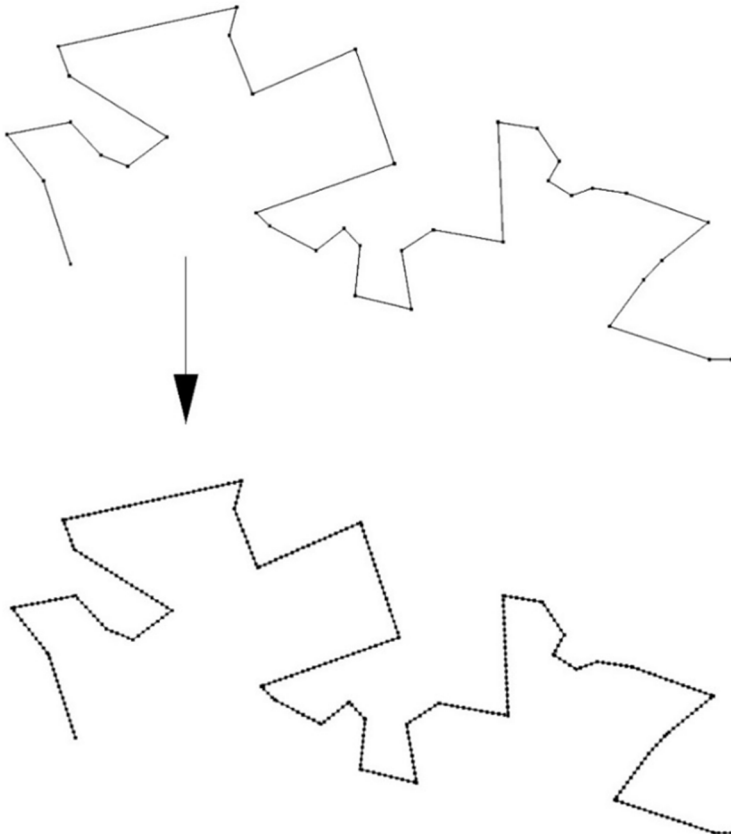


Figure 12. Converting smoothed polyline to dense and even samples

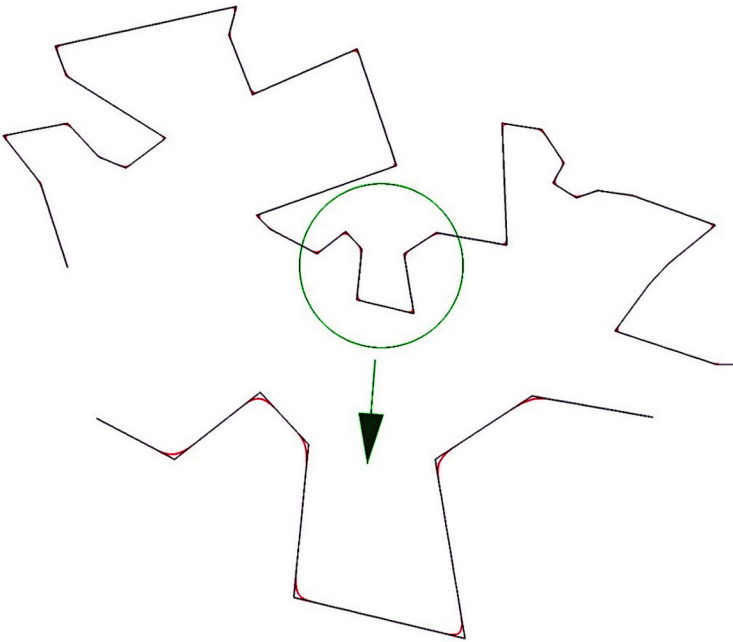


Figure 13. B-spline control point function, created on a set of samples, giving an initial, slight smoothing of the polyline

1. In the first stage, the set of vertices of the smoothed polyline is converted into a set of steps. The polyline is therefore sampled with a given *a priori* density. The higher the sampling density, the less initial smoothing can be achieved. Already with the number of samples being approximately 5 times the number of vertices of the input polyline, it is possible to obtain a curve that runs very close to the smoothed vertices. The curves presented below were generated for an initial number of samples that was 10 times the number of vertices, which allowed for detailed monitoring of subsequent smoothing iterations. The higher the sampling density, the less important it is for smoothing, whether the original vertices were added to the sample set, or whether it is just a set of equidistant points on the input polyline.

2. In the second stage, a B-spline control point curve is determined for the sampled set of points. This gives the initial, slightest smoothing of the polyline.

3. In the third stage, the previously determined B-spline function is sampled. To ensure sufficiently rapid progress in smoothing the

curves in subsequent iterations, the number of samples is reduced (this corresponds to the described increase in the hiker's steps). Good results, presented below, are achieved for approximately 10–20 percent reduction in the number of samples compared to the previous iteration.

4. In the next stages, further iterations are performed: each smoothed curve is sampled with decreasing density, and then a new B-spline function is constructed for the points sampled. If the process is unrestricted, smoothing is performed until a straight line is obtained between the starting and ending points of the input polyline. Smoothing can be carried out until the set tolerance of the distance of the smoothed curve from the input vertices is exceeded. The limit distance value can be set based on the target map scale.

The method is therefore based on standard B-spline control point functions and its effectiveness results from appropriate preparation and modification of input data in subsequent iterations, so as to remove the undesirable features of spline functions described in the previous section.

The following shows how this method will smooth the exemplary input polyline in subsequent iterations. In the first stage, the polyline spanning between the vertices is converted into a set of points located on the polyline, which corresponds to the hiker's steps (Figure 12). This set is dense and even.

Based on the set of samples, a B-spline control point function is created, giving an initial slight smoothing (Figure 13) on all points. This avoids the main drawback of these functions, which causes the functions to quickly move away from some vertices (Figures 7 and 8).

Then, the previously obtained B-spline function is sampled, and the number of samples is

smaller than the number of input polyline samples. The angles between the points marked on this curve are not so acute as those between the points of the input polyline. For this reason, the next B-spline curve will not exhibit an abrupt divergence from some vertices even though the points are less frequently spaced. A similar pattern will occur in each subsequent iteration. Each subsequent B-spline curve will be smoother and smoother, and the divergence from the previous one will be performed without sudden changes, which was visible in Figures 7 and 8. Figure 14 compares how the curve from the proposed method moves away from the vertices in relation to the B-spline

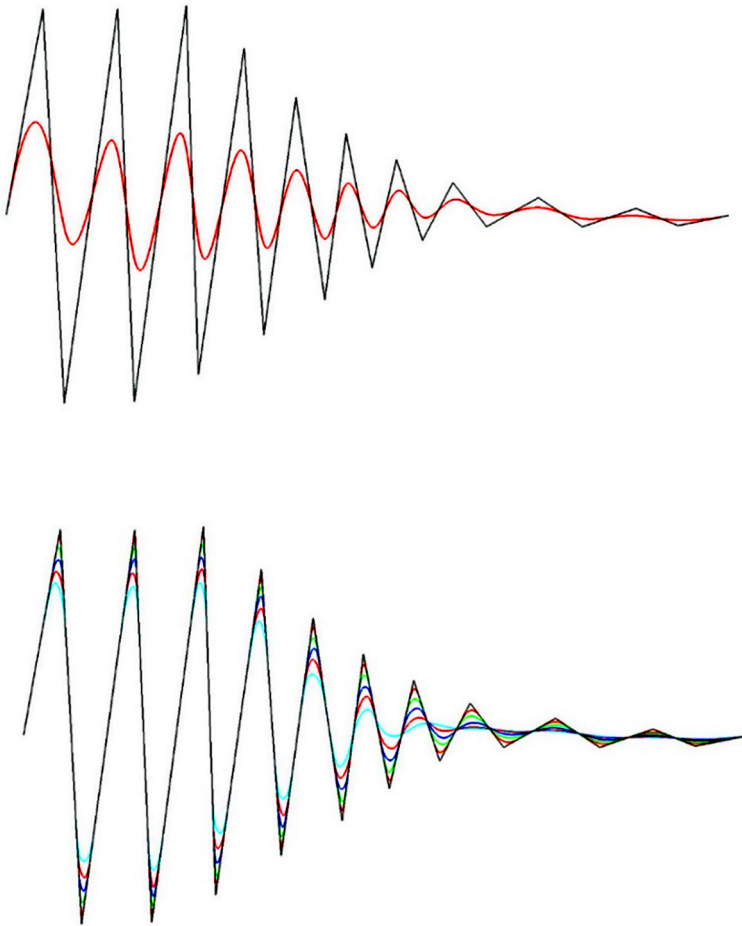


Figure 14. Comparison of divergence from the vertices. Top - B-spline control point function, based on vertices of a smoothed polyline. Bottom - B-spline control point functions, based on an iteratively reduced, dense set of equally spaced points for five successive iterations.

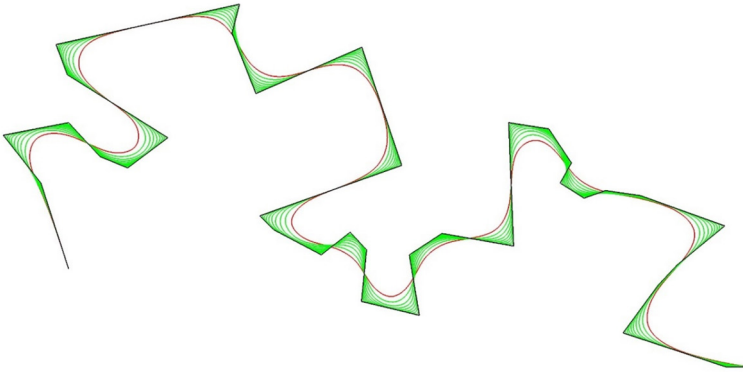


Figure 15. Smoothing curve (red) obtained in the ninth iteration based on the previous ones (green)

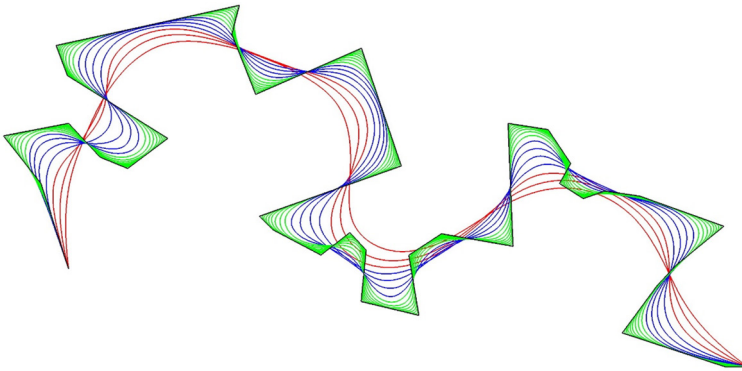


Figure 16. Smoothing curves that change colour when a given tolerance threshold is exceeded. Reduction of the number of samples in subsequent iterations by 20% compared to the previous iteration

function based directly on the vertices of the polyline.

Coming back to the analysed case of the polyline, e.g. after 9 iterations, for a set of samples reduced each time by 20% in relation to the number of samples of the previous iteration, a smoothed curve will appear (red), which is compared to the previous ones (green) in Figure 15. It can be noticed that the curve evenly smooths the input polyline. In initial iterations, the curve adheres well to all sections and the vertices are smoothed. In subsequent iterations, increasingly greater smoothing of the vertices, causes the smoothed segments to approach each other. The shorter the smoothed sections, the faster the smoothed sections approach, ultimately causing the curve to diverge from the straight section. The

smaller the irregularities of the input polyline, the faster they will be smoothed.

The curve never goes beyond the outline of the convex fragments of the polyline or any of the previous lines. With each iteration it becomes smoother, has less and less curvature and is shorter and shorter. Its shape becomes more and more simplified, to a borderline section based on the starting and ending points of the original curve. Iterations can be stopped earlier, i.e. when the set tolerance of the distance of the curve from the input polyline is exceeded. This tolerance may result from the ability to recognise details, which is related to the target map scale.

Figure 16 illustrates what smoothing will look like, assuming that the curve reaches three consecutive tolerance thresholds. After reaching

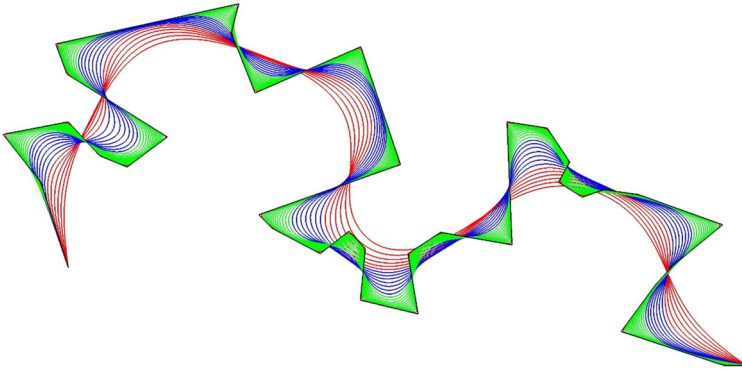


Figure 17. Smoothing curves that change colour when a given tolerance threshold is exceeded. Reduction of the number of samples in subsequent iterations by 10% compared to the previous iteration

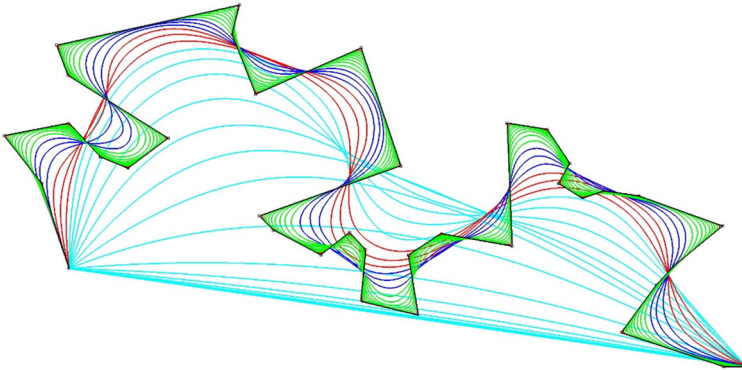


Figure 18. Smoothing without tolerance limits continues until a straight section is obtained. Reduction of the number of samples in subsequent iterations by 20%

each threshold, the colour of the curves changes in subsequent iterations. The last green curve is smoothed with a tolerance of one unit, the last blue one with a tolerance of two units, and the red one with a tolerance of three units.

Increasing distances between the curves of subsequent iterations are noticeable, which results from reducing the number of samples in each subsequent iteration. For Figure 16, the number of samples was reduced by 20% compared to the previous iteration. Figure 17 illustrates the smoothing curves with a sample reduction of 10%.

If the iterations are not interrupted by the given tolerance, smoothing will work until a straight line is obtained between the end points of the input polyline (Figure 18).

Figure 19 illustrates unlimited smoothing iterations for a more complex shape. From the main polyline, branches extend, which then divide into smaller segments. It can be observed that subsequent iterations smooth these small fragments and then, after reducing them, smooth the main branches.

The proposed method also works for closed curves. Since in this case there are no end points of the polyline, after smoothing the local kinks, the curve will subside towards the centre of the object in subsequent iterations (Figure 20).

Since the smoothing method combines two different but complementary elements: traversing the route using steps and smoothing using spline functions, the name *walking spline* was proposed for it.

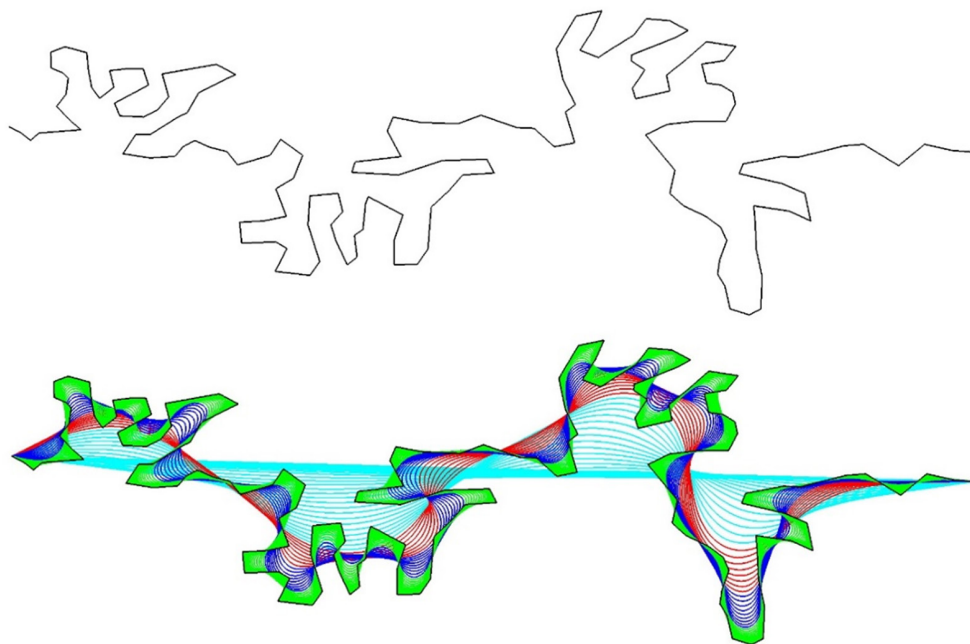


Figure 19. Smoothing a polyline with branches, which then divide into smaller branches. Reduction of the number of samples in subsequent iterations by 10%

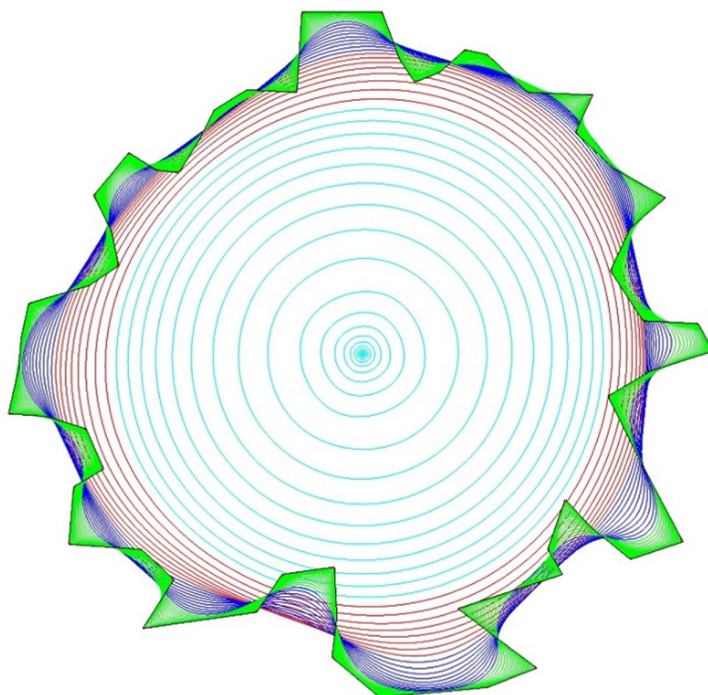


Figure 20. Smoothing a closed polyline. Reduction of the number of samples in subsequent iterations by 10%

6. Test Examples

The walking spline method was compared to the popular PAEK smoothing method, which is the standard among smoothing methods implemented in the ArcGis Pro software. Two different polylines were constructed, one with a lower (K1) and another with a higher (K2) degree of complexity. The polylines consist of a combination of several partial cases resulting from the varying lengths and slopes of the segments connecting an irregular set of points. The smoothing results for polyline K1 are presented in Figures 21–22, and for polyline K2 in Figures 23–25. The figures are accompanied by numerical analyses compiled in Table 1 for polyline K1 and Table 2 for polyline K2. They contain the value of the specified threshold deviation of the curve from the polyline and the calculated average deviation for it. The average deviation is derived from the area enclosed by the polyline and the smoothing curve, divided by the arc length of the smoothing curve.

It should be noted that line smoothing on maps is also assessed in terms of perception related to the map's readability. Achieving a smaller or larger maximum or average deviation does not necessarily determine better map readability. Therefore, the evaluation of line smoothing methods on maps is not based solely on numerical values (Bourghardt, 2005; Weibel, 2020; Saux, 2003; Guilbert & Lin, 2006; Podolskaia et al., 2013).

6.1. Smoothing the K1 curve

Figure 21a and Figure 22, being its enlarged fragment, show the operation of both methods for slight smoothing of the polyline *K1*. The results are very similar, but the *walking spline* method does not go beyond the outline of the convex fragments of the smoothed polyline. For greater smoothing of *K1* (Figures 21b and 21c), the results of both methods are comparable, and the verti-

Table 1. The threshold deviation of the smoothing curve from polyline K1, along with the calculated values for the area enclosed by the polyline and the curve, the arc length of the smoothing curve, and the average deviation

Model K1						
threshold deviation [m]	walking spline			PAEK		
	area [m ²]	curve length [m]	mean distance [m]	area [m ²]	curve length [m]	mean distance [m]
0.3	1.3	72.7	0.018	1.5	73.1	0.020
1.1	15.3	57.1	0.27	17.1	57.0	0.30
1.9	31.0	45.2	0.69	33.5	44.9	0.74
2.6	46.5	37.2	1.25	44.5	38.6	1.15
3.3	64.8	31.3	2.07	44.5	38.6	1.15
4.1	76.1	28.5	2.67	44.5	38.6	1.15
4.9	89.2	26.5	3.37	44.5	38.6	1.15
6.8	99.4	24.4	4.07	44.5	38.6	1.15
9.8	125.7	23.6	5.33	44.5	38.6	1.15

Table 2. The threshold deviation of the smoothing curve from polyline K2, along with the calculated values for the area enclosed by the polyline and the curve, the arc length of the smoothing curve, and the average deviation

Model K2						
threshold deviation [m]	walking spline			PAEK		
	area [m ²]	curve length [m]	mean distance [m]	area [m ²]	curve length [m]	mean distance [m]
0.1	0.5	116.7	0.004	0.6	117.4	0.005
0.5	10.4	91.2	0.11	11.8	90.7	0.13
1.2	23.6	64.8	0.36	25.4	64.2	0.40
1.8	32.8	49.8	0.66	34.6	49.5	0.70
2.8	41.2	38.9	1.06	46.2	37.5	1.23
4.0	47.5	34.3	1.38	46.2	37.5	1.23
5.1	51.9	31.6	1.64	46.2	37.5	1.23
5.4	54.8	31.1	1.76	46.2	37.5	1.23

ces are smoothed slightly more alternately by each method. In Figure 21d, maximum smoothing was achieved using the *PAEK* method, and further increases in tolerance no longer resulted in an increase in smoothing, as can be seen in Figures 21e–h. At the same time, the *walk-*

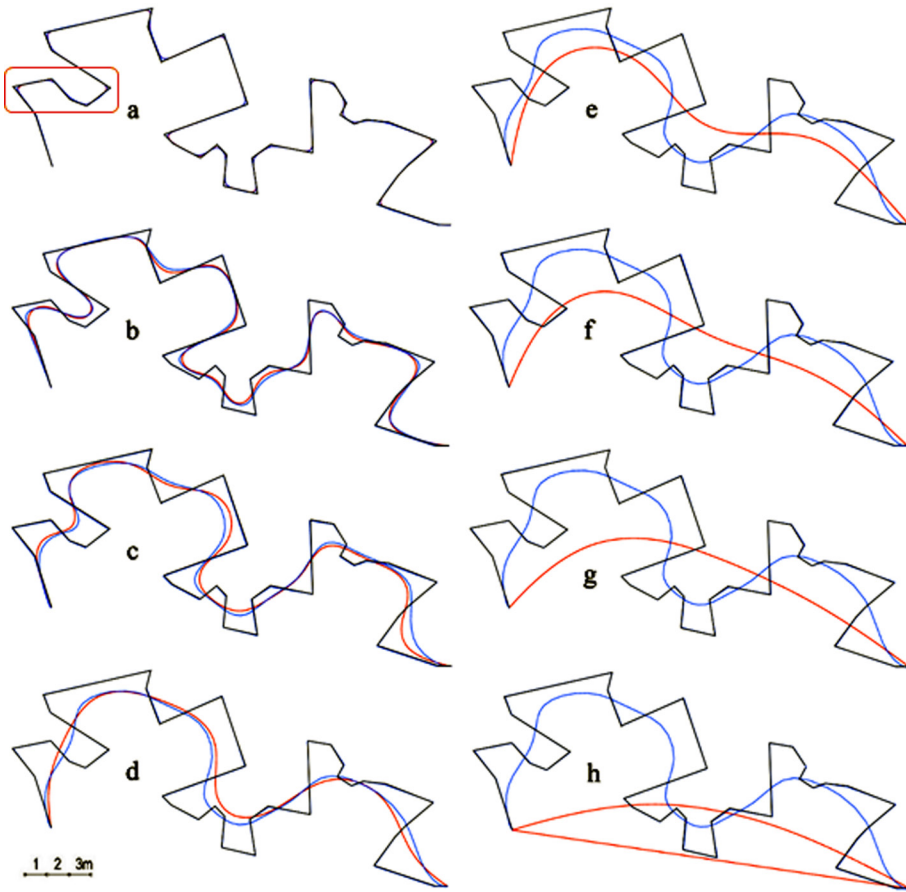


Figure 21. Smoothing of the polyline *K1* obtained using the *walking spline* method (red) compared to smoothing using the *PAEK* method (blue). Tolerances: a) 0.3m, b) 1.1m, c) 1.9m, d) 2.6m, e) 3.3m, f) 4.1m, g) 4.9m, h) 6.8m and 9.8m. The red frame in figure (a) marks the section that can be seen enlarged in figure 22

ing spline method, allows for obtaining an increasingly smoother curve compared to *PAEK*, and finally turns into a straight section in the last figure.

The numerical results confirm the observations derived from the figures. Additionally, they demonstrate that for a given value of threshold deviation, the walking spline method yields slightly smaller average deviations than the *PAEK* method. The lengths of the smoothed curve arcs are also very similar. Starting from a threshold deviation of 2.6 m, the *PAEK* method stops smoothing the curve, and the maximum achieved average deviation is 1.15 m. The walking spline method continues to smooth the polyline progressively with each subse-

quent iteration, and at the final threshold deviation of 9.8 m, the average deviation is 5.33 m. The reduction in arc length associated with smoothing for the *PAEK* method results in a final curve length of 38.6 m, whereas for the walking spline method, the curve length is 23.6 m.

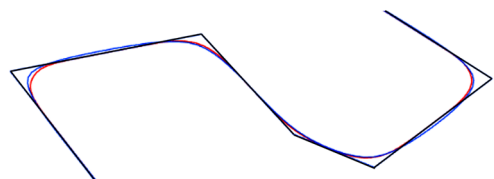


Figure 22. Enlarged fragment of Figure 21.a

6.2. Smoothing the K2 curve

A slight smoothing of polyline K2 is illustrated in Figure 23a and Figure 25, being its enlarged

fragment, obtaining results analogous to those for polyline K1. For increasing smoothing of the polyline K2, the *PAEK* and *walking spline* curves have a similar course (Figures 23b–d),

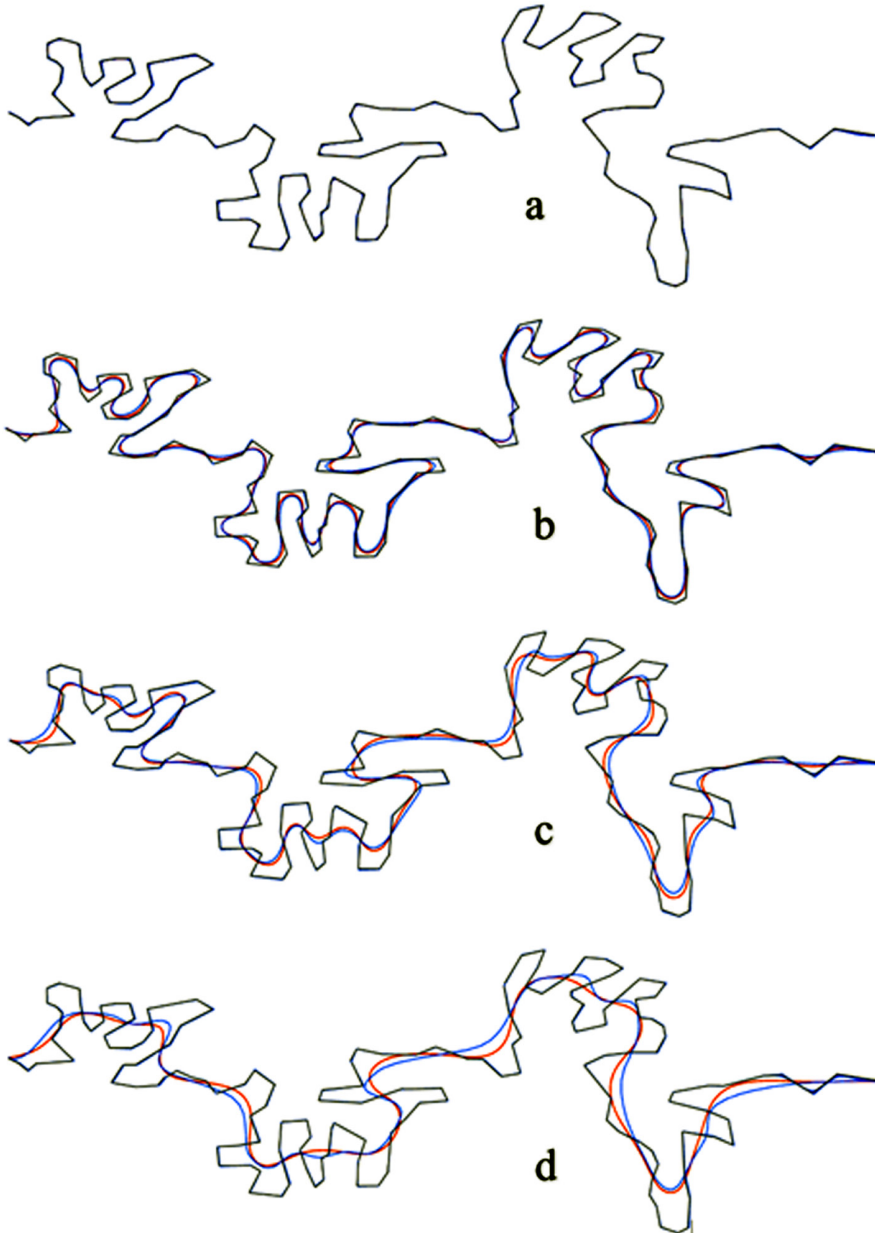


Figure 23. Smoothing of the polyline K2 obtained using the *walking spline* method (red) compared to smoothing using the *PAEK* method (blue). Tolerances: a) 0.1m, b) 0.5m, c) 1.2m, d) 1.8m. The red frame in figure (a) marks the section that can be seen enlarged in figure 25

but as the smoothing increases, differences begin to appear between them. Again, alternating, each method smooths some vertices slightly differently.

Maximum smoothing using the *PAEK* method, after which the increase in tolerance no longer changed the shape of the curve, was achieved for the polyline K2 in Figure 24a. The *PAEK* and

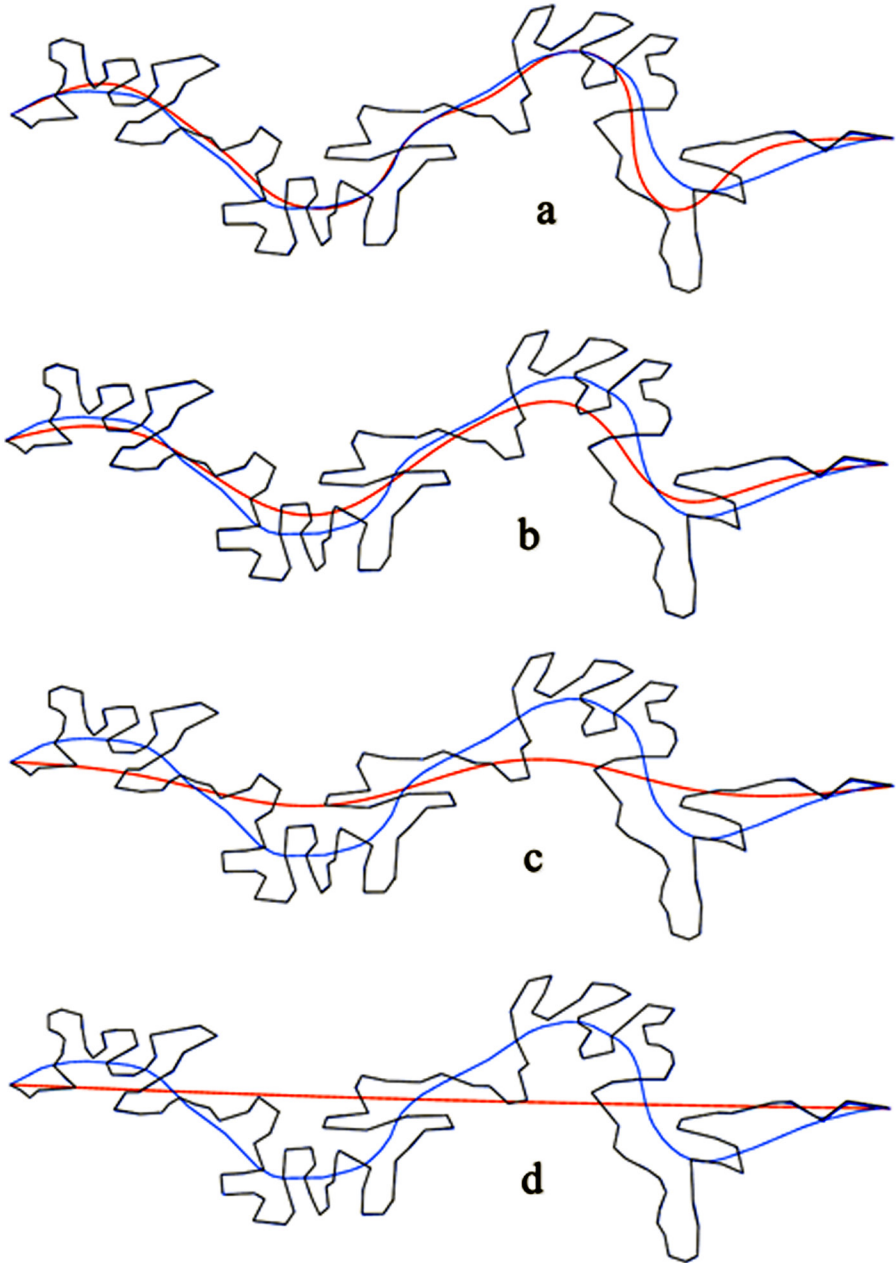


Figure 24. Smoothing the polyline K2 using the *walking spline* method (red) compared to smoothing with *PAEK* (blue). Tolerances: a) 2.8m, b) 4.0m, c) 5.1m, d) 5.4m

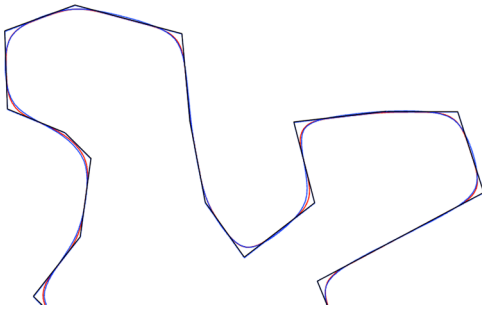


Figure 25. Enlarged fragment of Figure 23.a

walking spline curves provide similar smoothing, but at the end of the polyline, the *walking spline* curve seems to be better suited to it. Starting from Figure 24b, the *walking spline* curve becomes smoother and smoother compared to the no longer changing *PAEK* curve (Figure 24c, d) and finally turns into a straight section.

Similarly for polyline K1 with lower complexity, the *walking spline* method yields slightly smaller average deviations than the *PAEK* method for a given threshold deviation value. The lengths of the smoothed curve arcs are also comparable. Starting from a threshold deviation of 2.8 m, the *PAEK* method stops smoothing the curve, and the maximum achieved average deviation is 1.23 m. The *walking spline* method progressively smooths the polyline with each subsequent iteration, and at the final threshold deviation of 5.4 m, the average deviation is 1.76 m. The reduction in arc length associated with smoothing for the *PAEK* method results in a final curve length of 37.5 m, while for the *walking spline* method, the curve length is 31.5 m.

Generally, for small smoothing tolerances, the *walking spline* method yields results very similar to the *PAEK* method, but the curve does not extend beyond the outer contour of the convex fragments of the polyline. For larger

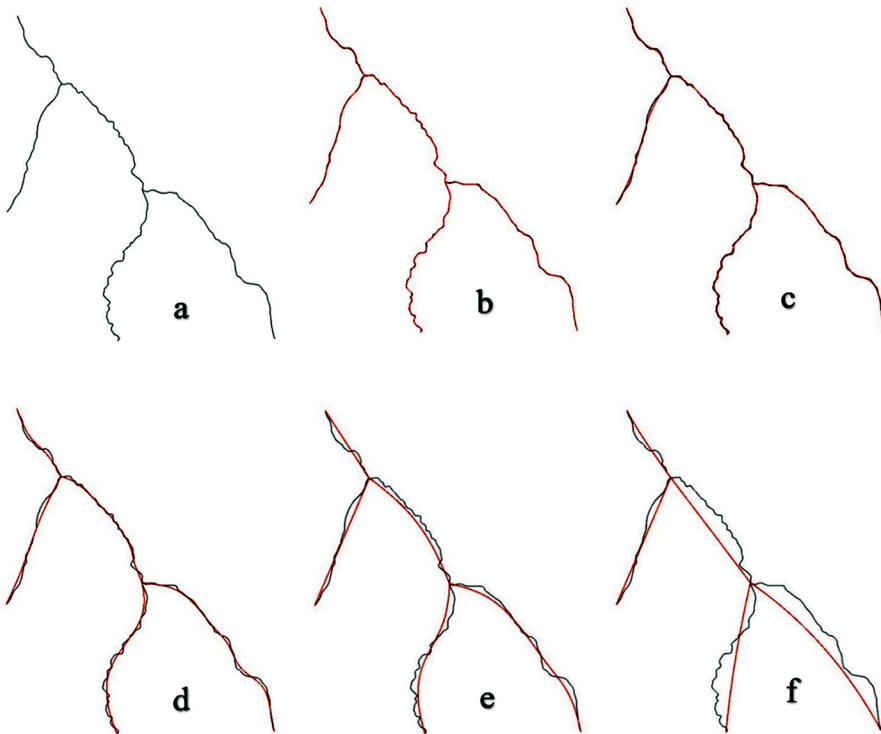


Figure 26. Example of smoothing a section of the Jąłowica river with its tributaries (a). Smoothing effects for the following scales: (b) 1:10,000, (c) 1:25,000, (d) 1:50,000, (e) 1:100,000, (f) 1:250,000, using the same scale for all drawings, so that the gradual smoothing process is visible

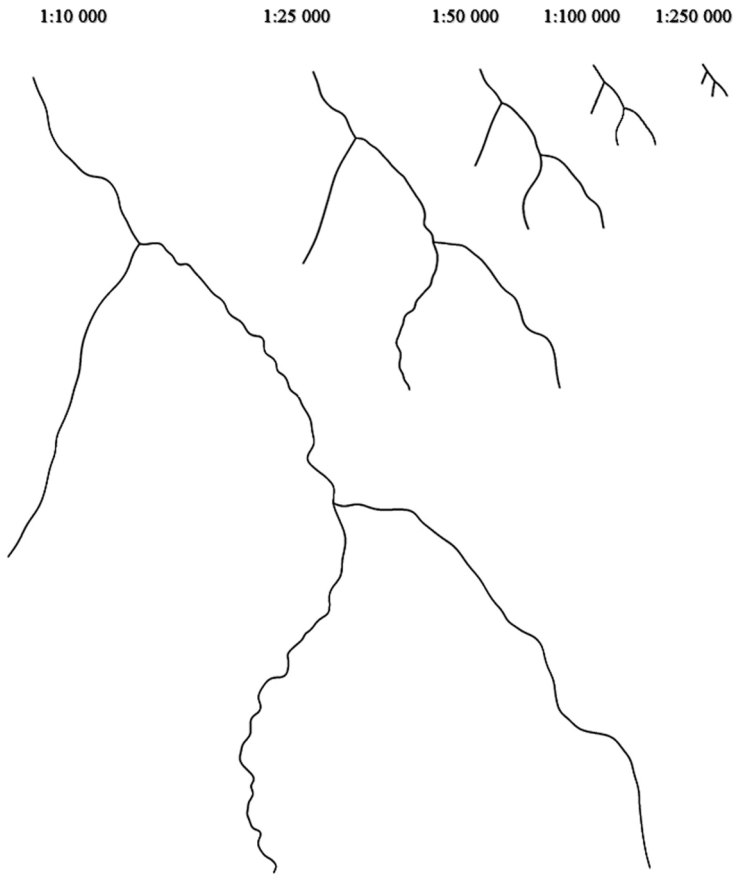


Figure 27. Example of smoothing a section of the Jalowica river with its tributaries at specified scales

smoothing tolerances, the results are similar, but each method alternately smooths some parts of the polyline differently. For large smoothing tolerances, the *PAEK* method stops the smoothing process, while the *walking spline* method gives an increasingly simplified course, until a straight line is obtained.

A comparison with the *PAEK* method is presented on generated data, which consists of shapes with high geometric variability, in order to demonstrate the effectiveness of the proposed method in complex cases. Below are examples of smoothing a section of the Jalowica river and its tributaries, which is a left tributary of the Biała Łądecka river, at scales of 1:10,000, 1:25,000, 1:50,000, 1:100,000 and 1:250,000. It was assumed that details smaller than 0.5 mm could be smoothed out on the

map scale. For the 1:10,000 scale, this will be details smaller than 5 metres, and for the other scales, analogously: 1:25,000 – smaller than 12.5 metres, 1:50,000 – smaller than 25 metres, 1:100,000 – smaller than 50 metres, 1:250,000 – smaller than 125 metres. Figure 26 shows the smoothing effects for all scales but using the same drawing scale so that the gradual smoothing process is visible. Figure 27 shows the smoothing effects at the specified scales. Characteristic points, i.e. the starting, ending and node points of the river and its tributaries, have been retained. Between these points, the lines are gradually smoothed until straight lines are formed between the characteristic points for less complex sections at a scale of 1:250,000.

7. Conclusions

The presented method combines a changed perception of the polyline on a map, with the use of smoothing using B-spline control point functions. The smoothed line is no longer just a line connecting a set of vertices on the map but becomes a path that the hiker must follow in the field. Traces left by dense and uniform steps constitute a data set, for which the unfavourable properties of spline functions are eliminated. Successive iterations creating curves, performed on an increasingly sparse, but still uniform set of traces, allow the polyline to be evenly smoothed. The B-spline curve diverges from the vertices evenly but remains “adhered” to the input polyline between them. The shorter the smoothed sections, the faster the smoothed fragments converge, ultimately causing the curve to diverge from the straight section. The smaller the irregularities of the input polyline, the faster they will be smoothed. Smoothing can be carried out until a given tolerance of the distance of the curve from the input vertices is exceeded, which may follow from

the target map scale. This may be the only parameter that determines smoothing. The second parameter may be the degree of data reduction for each subsequent iteration, affecting the iteration density and thus the increment of the smoothing value. Each subsequent curve becomes smoother and shorter, simplifying the input polyline until a straight line is obtained between its ends. For closed curves, after smoothing out local irregularities, the proposed method will create curves that subside towards the centre of the figure.

The proposed polyline smoothing method is aimed, among others, at geoinformaticians who prepare online maps and geovisualizations using GIS/CAD tools. The method can meet specific requirements for simplifying the shapes of linear objects on online-scaled maps: shape similarity and, in the case of maps at larger scales, the cartometric accuracy of the drawing.

Acknowledgements. This work was supported by the research subvention of AGH University of Science and Technology No. 16.16.150.545 in 2025.

References

- Ahlberg, J. H., Nilson, E. N., & Walsh, J.L. (1967). The theory of splines and their applications. Academic Press Inc., New York.
- Bac-Bronowicz, J., Banasik, P. & Chrobak, T. (2023). Automatic simplification of the geometry of a cartographic line using contractive self-mapping – illustrated with an example of a polyline band. *Polish Cartographical Review*, 55, 73–86. <https://doi.org/10.2478/pcr-2023-0007>
- Baig, S., Rahman, A., & Duncan, E. (2013). A review and conceptual framework for generalization of maps. In Developments in multidimensional spatial data models: Lecture notes in geoinformation and cartography. https://doi.org/10.1007/978-3-642-36379-5_12
- Bayer, T., Kolingerová, I., Čelonk, M., & Lysák, J. (2023). Simplification of contour lines, based on axial splines, with high-quality results. *International Journal of Geographical Information Science*, 37(7), 1520–1554. <https://doi.org/10.1080/13658816.2023.2193969>
- Birkhoff, G., & de Boor, C. (1964). Error bounds of spline interpolation. *Journal of Mathematics and Mechanics*, 13, 827–835.
- Bodansky, E., Gribov, A., & Pilouk, M. (2002). Smoothing and compression of lines obtained from raster-to-vector conversion. In D. Blostein, & Y. B. Kwon (Eds.), Graphics recognition algorithms and applications: GREC 2001 (pp. 235–243). Springer. https://doi.org/10.1007/3-540-45868-9_22
- Brigger, P., Hoeg, J., Unser, M. (2000). B-spline snakes: a flexible tool for parametric contour detection. *IEEE Transactions on Image Processing*, 9(9), 1484–1496. <https://doi.org/10.1109/83.862624>
- Burghardt, D. (2005). Controlled line smoothing by snakes. *Geoinformatica*, 9, 237–252. <https://doi.org/10.1007/s10707-005-1283-3>
- Cebrykow, P. (2017). Cartographic generalization yesterday and today. *Polish Cartographical Review*, 49, 5–15. <https://doi.org/10.1515/pcr-2017-0001>
- de Boor, C. (1978). *A practical guide to splines*. Springer-Verlag, New York.
- Farin, G. (2002). Curves and surfaces for computer-aided geometric design. Academic Press. <https://doi.org/10.1016/B978-012249048-5/50010-3>
- Floater, M.S., & Surazhsky, T. (2006). Parameterization for curve interpolation. *Studies in Computational Mathematics*, 12, 39–54. [https://doi.org/10.1016/S1570-579X\(06\)80004-2](https://doi.org/10.1016/S1570-579X(06)80004-2)
- Foerster, T., Stoter, J., & Köbben, B. (2007). Towards a formal classification of generalization operators. In Proceedings of the 23rd International Carto-

- graphic Conference (Vol. 4). International Cartographic Association.
- Guilbert, E., & Lin, H. (2006). B-spline curve smoothing under position constraints for line generalisation. In Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems (3–10). ACM. <https://doi.org/10.1145/1183471.1183474>
- Guilbert, E., & Saux, E. (2008). Cartographic generalisation of lines based on a B-spline snake model. *International Journal of Geographical Information Science*, 22, 847–870. <https://doi.org/10.1080/13658810701689846>
- Guo, Q., & Zhou, L., & Wang, L., & Sun, Y., & Li, X. (2017). Improvement of Snake Displacement Model for Roads Considering Cartographic Rules. *Geomatics and Information Science of Wuhan University*, 42, 1629–1634. <https://doi.org/10.13203/j.whugis20160357>
- Haron, H., Rehman, A., Adi, D., Saba, T., & Lim, S. P. (2012). Parametrization method on B-spline curve. *Mathematical Problems in Engineering*, 2012, Article 640472. <https://doi.org/10.1155/2012/640472>
- Heckbert, P., & Garland, M. (1997). Survey of polygonal surface simplification algorithms (Technical report). School of computer science, Carnegie Mellon University.
- Jiang, B., Xu, S., & Li, Z. (2023). Polyline simplification using a region proposal network integrating raster and vector features. *GIScience and Remote Sensing*, 60(1). <https://doi.org/10.1080/15481603.2023.2275427>
- Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: active contour models. *International Journal of Computer Vision*, 1, 321–331.
- Kettunen, P., Koski, C., & Oksanen, J. (2017). A design of contour generation for topographic maps with adaptive DEM smoothing. *International Journal of Cartography*, 3(1), 19–30. <https://doi.org/10.1080/23729333.2017.1300998>
- Kiciak, P. (2019). Podstawy modelowania krzywych i powierzchni: zastosowania w grafice komputerowej. [Fundamentals of curve and surface modeling: applications in computer graphics]. Wydawnictwo Naukowe PWN, Warszawa.
- Kuna J., Jeremicz J., Kociuba D., Niedźwiadek, R., Janus, K., Chachaj, J. (2024). The challenges of reconstructing the historic urban landscape of Lublin in the Lublin Union period (1569) in an interactive map. *Studia Geohistorica*, 12, 151–198. <https://doi.org/10.12775/SG.2024.08>.
- Li, Z. (2007a). Digital map generalization at the age of enlightenment: A review of the first forty years. *The Cartographic Journal*, 44, 80–93. <https://doi.org/10.1179/000870407X173913>
- Li, Z. (2007b). Algorithmic foundations of multi-scale spatial representation. CRC Press. <https://doi.org/10.1201/9781420055435>
- Li, H., Li, Z., & Mo, W. (2017). A time varying filter approach for empirical mode decomposition. *Signal Processing*, 138, 146–158. <https://doi.org/10.1016/j.sigpro.2017.03.019>
- Luebke, D. P. (2001). A developer’s survey of polygonal simplification algorithms. *IEEE Computer Graphics and Applications*, 21(7), 24–35. <https://doi.org/10.1109/38.920624>.
- McMaster, R.B., & Shea, K.S. (1992). Generalization in digital cartography. Association of American Geographers. Washington.
- Muller, J. C. (1991). Generalization of spatial databases. In D. J. Maguire, M. F. Goodchild, & D. W. Rhind (Eds.), *Geographical information systems: Principles and applications* (Vol. 1, pp. 457–475). Longman.
- Neun, M., Burghardt, D., & Weibel, R. (2009). Automated processing for map generalization using web services. *Geoinformatica*, 13, 425–452. <https://doi.org/10.1007/s10707-008-0054-3>
- Nieuwenhuizen, N., Lindsay, J. B., & DeVries, B. (2021). Smoothing of digital elevation models and the alteration of overland flow path length distributions. *Hydrological Processes*, 35(7), Article e14271. <https://doi.org/10.1002/hyp.14271>.
- Nöllenburg, M., Merrick, D., Wolff, A., & Benkert, M. (2008). Morphing polylines: A step towards continuous generalization. *Computers, Environment and Urban Systems*, 32, 248–260. <https://doi.org/10.1016/j.compenvurbsys.2008.06.004>.
- Esri (2024). *How smooth line and smooth polygon work*. <https://pro.arcgis.com/en/pro-app/3.1/tool-reference/cartography/how-smooth-line-and-smooth-polygon-work.htm>
- Peterson, J. W. (2006). Arc length parameterization of spline curves. Taligent, Inc. <https://www.saccade.com/writing/graphics/RE-PARAM.PDF>, 1–11.
- Piegl, L., & Tiller W. (2012). The NURBS book. Springer-Verlag, Berlin.
- Podolskaia, E., Anders, K.-H., Haurert, J.-H. & Sester, M. (2013). Quality assessment for polygon generalization. <https://doi.org/10.1201/9781420069273.ch16>
- Saux, E. (2003). B-spline functions and wavelets for cartographic line generalization. *Cartography and Geographic Information Science*, 30(1), 33–50. <https://doi.org/10.1559/152304003100010938>
- Schumaker, L. (2007). Spline functions: Basic theory. Cambridge University Press.
- Shan, B., Ni, W., Yuan X., Yang, D., Wang, X., & Liu, R. P. (2023). Graph learning from band-limited data by graph Fourier transform analysis. *Signal Processing*, 207, Article 108950. <https://doi.org/10.1016/j.sigpro.2023.108950>
- Spoerhase, J., Storandt, S., & Zink, J. (2019). Simplification of polyline bundles. In Proceedings of the 35th International Symposium on Computational Geometry (SoCG ’19) (pp. 1–16). ACM. <https://doi.org/10.48550/arXiv.1907.05296>

- Steiniger, S., & Meier, S. (2004). Snakes: A technique for line smoothing and displacement in map generalisation. In 7th ICA Workshop on Generalisation and Multiple Representation (pp. 1–8). International Cartographic Association.
- Tutić, D., & Lapaine, M. (2010). New method for reducing sharp corners in cartographic lines with area preservation property. In Proceedings of the 14th International Conference on Geometry and Graphics (pp. 289–290). International Society for Geometry and Graphics.
- Weibel, R., & Jones, C.B. (1998). Computational perspectives on map generalization. *Geoinformatica*, 2, 307–314. <https://doi.org/10.1023/A:1009748903798>
- Weibel, R. (2020). Three essential building blocks for automated generalization. In L. Christophe, S. Mustière, & J. C. Ruas (Eds.), *Generalisation in the age of AI: Matching human intuition with machine learning* (pp. 56–69). CRC Press. <https://doi.org/10.1201/9781003062646-7>
- Velut, J., Benoit-Cattin, H., & Odet, C. (2007). Locally regularized smoothing B-snake. *EURASIP Journal on Advances in Signal Processing*, 2007, Article 76241, 1–12. doi: <https://doi.org/10.1155/2007/76241>
- Zaksek, K., & Podobnikar, T. (2005). An effective DEM generalization with basic GIS operations. In Proceedings of the 8th ICA Workshop on Generalisation and Multiple Representation. International Cartographic Association. https://gitlab.com/ica-gen/workshop-proceedings/-/raw/main/ica-gen-downloads/ica-gen/workshop2005/Zaksek_Podobnikar.pdf