AN ALGORITHM FOR AUTOMATED DIGITAL ROCK DRAWING IN THE STYLE USED IN CZECH TOPOGRAPHIC MAPS

JAKUB LYSÁK

Charles University in Prague, Faculty of Science, Department of Applied Geoinformatics and Cartography, Czech Republic

ABSTRACT

The Land Survey Office of the Czech Republic developed a method for digital cliff drawing for use in the production of their large-scale topographic maps. At the core of this method is the process of filling a polygon with lines that resemble stylized hachures. As these lines are produced manually, this article aims at the automation of this process, and presents an algorithm for automated digital rock drawing. This algorithm was designed based on a previous detailed analysis of the problem, and experience supported by numerous examples found in maps. The individual steps of this algorithm, which was successfully tested on selected areas of various rocky terrains in the Czech Republic, are described in detail. The results of the tests are assessed, and the pros, cons and limitations of the algorithm are discussed.

Keywords: digital cartography, digital cliff drawing, rock hachures, topographic map

Received 31 October 2014; Accepted 12 November 2015

1. Introduction

Digital cartography has greatly expedited and simplified the process of producing maps. However, digital cartography technology has not resolved all complications faced by the practice. This also applies to the production of rock hachures, a traditional and widely-used means for the cartographic representation of rocky terrain (Imhof 1965). Within the industry, such depictions have been developed to near perfection, particularly in Switzerland. On the other hand, drawing Swiss-style rock hachures requires a great deal of time and experience, and cannot be greatly assisted by digital processing (Gilgen, Jenny 2010). Nevertheless, this method remains an excellent technique for depicting high-mountain relief. Because such depictions are hand-drawn, even using a computer is time-consuming, and therefore expensive. As a result, researchers are seeking ways to automate the production process. Despite partial success, this still remains an issue.

This article focuses on a similar problem that affects another style of rock drawing. Cartographers from the Land Survey Office, the Czech national mapping agency, developed a way to manually produce digital drawings of rock formations. The basic principles of this technique were briefly summarised in the first section of this article. More detailed information about the representation and its components can be found in Lysák (2015). Existing work on the automation of digital rock drawing is presented below; however, they concern other styles of cartographic representation.

The ultimate goal of this work is to design an algorithm for an automatic cartographic representation of rocky terrains in the style used by the Land Survey Office of the Czech Republic in their topographic maps (example in Figure 1). A description of the procedure includes the requirements of its input data, an in-depth explanation of the crucial processing steps, a comparison with manual processing, a discussion of the experimental implementation in ArcGIS for Desktop software, and the results of tests on real data. The proposed algorithm and its experimental implementation should not only help cartographers, but also researchers dealing with large-scale mapping of rocky terrains, such as geomorphologists and geologists. Therein should lie the practical relevance of the outcomes of this research.

2. The style of digital rock drawing used in Czech topographic maps

As this work immediately follows from a previous article by Lysák (2015) discussing this style of digital rock drawing, the description here is brief and limited to the basic principles. More information can be found in the cited article. An example of this type of portrayal can be seen in Figure 1.

The method is based on filling a polygon with lines. First, symbols for a single rock are created, resembling a stylised hachure. These are placed sequentially along the upper edge of a rock (like pearls on a necklace). The symbols (referred to as "upper symbols") change regularly; the smaller ones serve as transitional strokes between the larger ones, which help to avoid to some extent undesirable regularity. There are also variants for wide and narrow rocks.

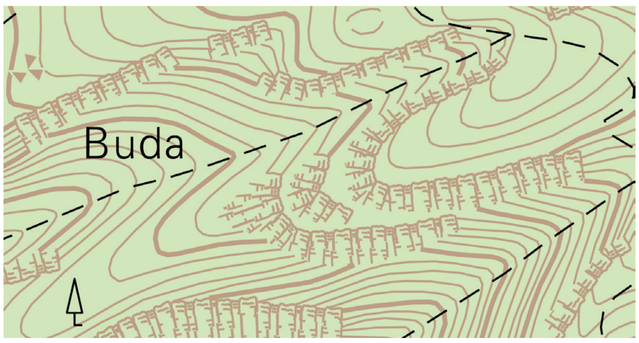

Fig. 1 An example of a digital rock drawing using the described method. Map taken from Geoportal ČÚZK, © ČÚZK.

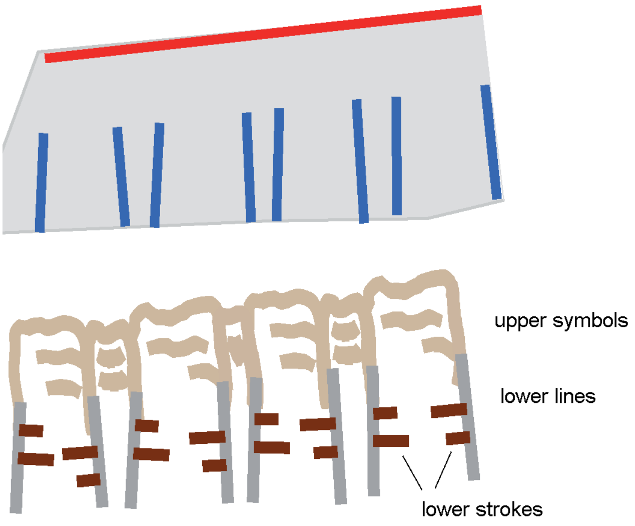

Fig. 2 From lines to hachures. The upper figure portrays the polygon geometry from ZABAGED with lines that needed to be drawn by a cartographer. The lower figure illustrates the resulting representation. To illustrate more clearly, single components of the result are distinguished by colour. Data from Geoportal ČÚZK, © ČÚZK.

For the remaining portion of the polygon, lines depicting the lower part of a rock are utilised (referred to as "lower lines"). These lines generally follow the fall direction and have regularly distributed, short, transverse strokes (referred to as "lower strokes"). These lines are drawn to fill the part of the polygon that is not covered by the upper symbols, and meet them precisely. The process is illustrated in Figure 2. Lower lines are used in the case of larger (and especially wider) polygons. All lines are drawn manually by a cartographer.

3. Existing work on the algorithmization of digital rock drawings

A thorough inspection of existing work concerning rock representation and digital cartography was performed. Although this is not a very active area of cartographic research, digital cliff drawing has been studied, especially in Switzerland at ETH Zürich. The pioneer works in this field are summarised in Hurni, Dahinden, Hutzler (2001). The article contains an in-depth description of digital ridge-line representations and the details of its implementation, as well as attempts to generate shadow hachures used in Swiss-style maps. Further studies of different aspects of digital cliff drawing in Swiss-style were introduced (Dahinden, Hurni 2007; Gilgen 2007; Gilgen, Jenny 2010; Geisthövel 2013). Work on the automation of this type of rock drawing is ongoing and, despite a great deal of effort, there seems to be a long way from the realisation of a fully working solution. Unlike the method used in Czech topographic maps, this "Swiss Manner" produces excellent results and unparalleled representations of three-dimensionality. However, this is based on complicated rules that prove demanding, even for manual depiction, and appear not to be easily algorithmizable (Jenny et al. 2014). Nevertheless, one promising work in this field has been published recently (Geisthövel, Hurni 2015) which combines cartography and advanced algorithms of digital image processing and non-photorealistic rendering.

Articles by authors outside of Switzerland should also be mentioned, despite being scarce: Gondol, Le Bris, Lecordix (2008) focused on automation using oriented hatched patterns; or simulating rock textures using special digital terrain model (DTM) filtering called texture shading described in (Brown 2014), and Yang, Guo, Shen (2009), whose results are a bit closer to the Czech-style representation used on analogue topographic maps. The results of research on slope hachures in general are also particularly useful. Of these, Yoeli (1985) and Regnauld, Mackaness, and Hart (2002) are noteworthy.

From a general point of view, two approaches for creating hachures can be used: one is based mostly on the analysis of raster DTM using algorithms from raster GIS or digital image processing (image filtering, feature extraction, etc.), whereas the second one relies on vector features as an input and tries to perform some kind of "automatic drawing" using algorithms related to computer graphics or computational geometry. The algorithm proposed in this article more likely belongs to the second group, although it also uses information derived from a DTM.

Most of the papers mentioned above aim to design a representation of the large rock masses in high mountains and in their local-specific style; thus they are only applicable to local Czech conditions to a limited extent. With this in mind, we come to another approach described in this article, based on different rules, but leading to a solution to the same problem.

It should be also mentioned in this context that for many practitioners, rock hachures seem to be an anachronism in the era of GIS and detailed DTMs. They also justifiably ask whether it makes sense to put so much effort into the algorithmization of something that is not necessarily needed anymore. The dispute about hachures versus contours is long-standing. Arguments from the pre-GIS period for and against both methods are summarized in Imhof (1965) in the chapter *Critical examination and application of the different methods of rock drawing*. In the Czech Republic, another purely practical problem has appeared. In the time of the first digital processing of topographic maps based on the vectorization of their scanned analogue versions, neither contours nor a DTM was available in rocky areas, expressed with hachures (Lysák 2015). Thus, something needed to be drawn inside the polygons. Despite the fact that DTM from laser scanning data is now available for the entire territory of the Czech Republic (including rocky areas), the same argument can concern other countries, where large-scale topographic maps are only available for detailed spatial information. On the other hand, the idea that the time of a cartographer can be spent more efficiently than in the manual hand-drawing of hachures to reach aesthetic perfection in a map, seems to be generally reasonable. Therefore, the automatic production of rock hachures appears to be the only way to prevent this unique type of portrayal from being consigned to cartographic history.

4. Algorithm for automatic processing

The process of filling a polygon with hachures, based on the principles described in Section 2, is a rather time-consuming task. On the other hand, the individual steps seem to be clear and simple enough to be algorithmized. Generally, details of the rules were inferred from in-depth examinations of rock representations on maps produced by the Land Survey Office, and in consultation with experienced cartographers from the same institution.

4.1 Inputs and outputs of the proposed algorithm

The key input for processing is a polygon layer, representing an extent of the rocky area into which hachures will be drawn. Lines representing details of the rocky terrain inside these polygons can also be used as supporting information. The algorithm can deal with ridge lines, valley lines, and other terrain edges. Delineation of the outer limits of rocky terrain and extraction of relief features can, in some cases and to some extent, be automated. However, these issues are beyond the scope of this paper. The described process assumes the layers mentioned above are present, regardless of how they were created.

A DTM is essential for the classification of a polygon perimeter (described in Section 5.1). Extensive detail is not necessary as it is used only to distinguish between the upper and lower part of a rock object, allowing for the correct orientation of stylized hachures. The level of detail in the DTM should be comparable to or higher than the level of detail of a polygon delimitating the extent of rocky areas; otherwise the results of automatic classification can be very poor. DTMs interpolated from contour lines, lidar or radar data can be used. If a DTM is not available, classification must be done manually, although the other steps can still be automated.

Output data is a set of polylines consisting of upper symbols, lower lines and lower strokes, as defined in Section 2 and shown in Figure 2. Individual phases of the process also have their partial outputs, serving as inputs for the following phases. They can be edited by the user if they do not meet their needs. These phases are described in detail in the following paragraphs.

4.2 Overall description of the algorithm

In the following sections, we describe the design of the algorithm and discuss thoroughly the key steps involved. This was not necessarily aimed at a fully automated solution. The proposed process allows for user interaction in certain phases of processing, in terms of manual editing of intermediate results or correction of minor errors, which seems to be inevitable, even in a complicated terrain. The process should handle the most typical and the most frequent cases, though not necessarily everything, as manual intervention is often a faster and more reliable way to achieve the desired results. In addition, several enhancements to the cartographic representation itself are suggested.

5. Processing steps

The proposed algorithm starts with an analysis of the terrain, followed by checking the results and determination of which input polygons can be processed fully automatically. Each polygon is split into two parts: the lower part is decorated with lower lines and lower strokes, and the upper part is filled with upper symbols. Finally, some improvements to the aesthetic quality of the output are applied. The following text gives an analysis of the studied problem, describing individual phases of the process in Sections 5.1–5.7, and explaining key parameters, techniques and processing alternatives.

5.1 Classification of the polygon perimeter

The following process requires a distinction between the "upper" and "lower" parts of rock, as the resulting symbolization strongly depends on this. More formally, if the perimeter of a polygon is supposed to consist of line segments, each segment will be classified as "upper", "lower", "indefinite" or "conflicting". Because the polyline is a border of the polygon, each segment has the interior of the polygon on one side, and the exterior on the other. Further, the direction of the steepest descent for both the startpoint and endpoint of each segment is computed (referred to as "fall line vector" in this text). A vertex gets a + 1 mark if the corresponding fall line vector lies in the same half-plane as the interior of a polygon, a − 1 mark if it lies in the same half-plane as the exterior of a polygon, and a 0 mark if it lies exactly on or very close to the border (see Figure 3, top). This can be easily performed by a half-plane test (de Berg et al. 2008). The mark is related to the endpoints of the segment, i.e. the same point shared by adjacent segments and can have different marks in each segment. The *alpha* angle between the fall line vector and the direction vector of a line segment was also computed. This value was used to differentiate between "0" and "+/−1" vertices, using a threshold of *t* (if *alpha* < *t*, vertex gets a 0 mark).

For computation of a fall line vector, a DTM is needed. For the given point P = [*x*, *y*], a fall line vector f = (*dx*, *dy*) using the following formulae can be calculated, based on a derivative of bilinear interpolation of four points:
$dx = -1/s \cdot (-\mathrm{dtm}[x - s, y - s] + \mathrm{dtm}[x - s, y + s] - \mathrm{dtm}[x + s, y - s] + \mathrm{dtm}[x + s, y + s])$,
$dy = -1/s \cdot (-\mathrm{dtm}[x - s, y - s] - \mathrm{dtm}[x - s, y + s] + \mathrm{dtm}[x + s, y - s] + \mathrm{dtm}[x + s, y + s])$,
where dtm[*a*, *b*] is the *z*-coordinate at the position (*a*, *b*) and *s* (sampling) is a distance of points used for interpolation. The lower the value of *s* is the more terrain details are taken into account. The higher it is, the more a global perspective is used. The value of *s* must be set carefully, taking into consideration both the grid size of the DTM and the level of detail of the polygon data (i.e. the desired scale of cartographic representation). Tuning the value of *s* is advantageous as it helps to avoid undesirable details in the DTM, which might otherwise interfere with classification.

Based on marks of its startpoint and endpoint, segments can be classified, using the following rules (cf. Figure 3, bottom):

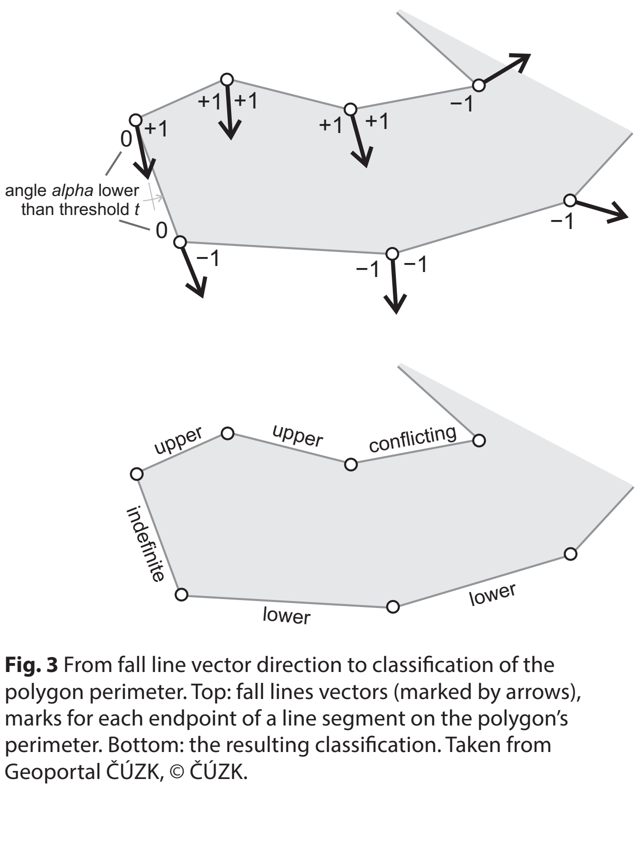

Fig. 3 From fall line vector direction to classification of the polygon perimeter. Top: fall lines vectors (marked by arrows), marks for each endpoint of a line segment on the polygon's perimeter. Bottom: the resulting classification. Taken from Geoportal ČÚZK, © ČÚZK.

– upper, if both vertices have +1 marks or if one has +1 and the other 0,
– lower, if both vertices have −1 marks or if one has −1 and the other has 0,
– indefinite, if both vertices have 0 marks,
– conflicting, if otherwise (one vertex has +1 and the other has −1).

Obviously, the higher the value of the *t* threshold, the more reliable the results obtained, in terms of minimizing the number of erroneously marked vertices. On the other hand, the more vertices that get a 0 mark, the more segments that will be indefinite. To minimize the number of conflicting segments, longer segments should be split. The presence of conflicting segments indicates a discrepancy between the DTM and the delineation of a rocky terrain. This can be solved to some extent by increasing the *t*, i.e. allowing small position errors, resulting in more indefinite segments.

For this reason, the generalization of segment classification is a reasonable step. In this case, generalization means applying simple rules that eliminate conflicting and some indefinite segments. The rules are as follows:
– dissolve segments of the same type, i.e. adjacent segments of the same type become connected and form a single dissolved polyline of a certain type,
– select indefinite and conflicting dissolved polylines, which are connected with an upper dissolved polyline on both sides, or with a lower dissolved polyline on both sides, and both polylines are long enough in comparison with the former ones. These dissolved polylines are reclassified as lower or upper (depending on the type of adjacent lines).

All lower and upper dissolved polylines and all indefinite lines that connect upper and lower polylines (or in other words, the "side parts", i.e. not the upper or lower part of a rock), are considered to be processed correctly. Classification of the remaining indefinite and conflicting parts must be done manually by the user. If the DTM is not reliable enough or not available, the classification of segments can be also performed completely by hand.

5.2 Verifying a polygon

As hachures from the upper to the lower part of a polygon need to be drawn, the natural requirement is that every polygon must have parts on its perimeter classified as lower and upper. In a simple case, the polygon has exactly one upper polyline, one lower polyline, and no more than two indefinite polylines. In this text, this type is called a "simple polygon", and a polygon with more than one upper polyline or more than one lower polyline is a "complex polygon". Complex polygons can be split into parts so that each part has the characteristics of a simple polygon. This process has yet to be automated.

If the polygon completely lacks lower or upper polylines, it indicates a need for information on the inner

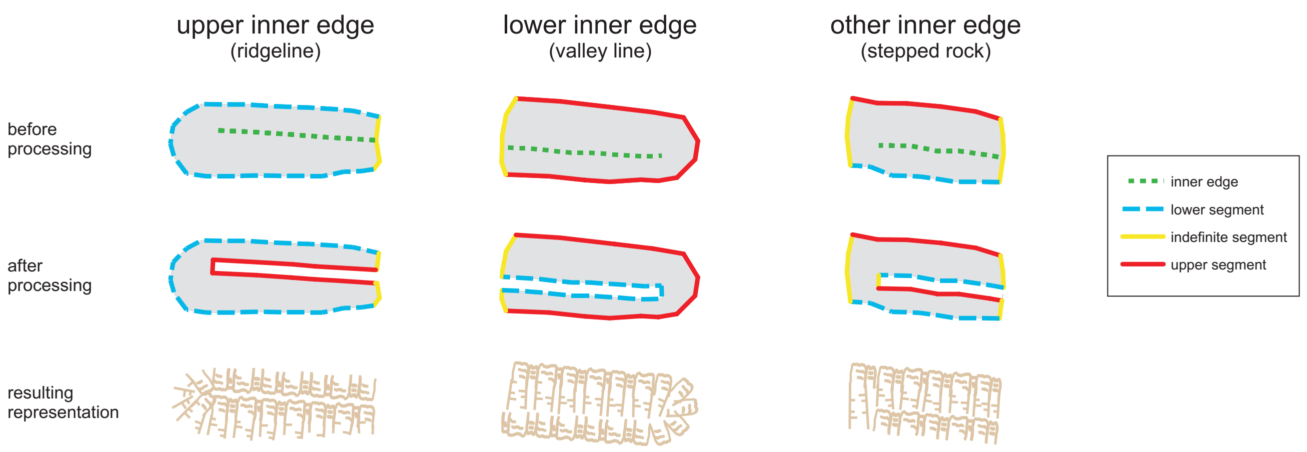

Fig. 4 Incorporation of inner edges. First row: classified segments on the perimeter of a polygon. Second row: the result after processing the edges. Third row: the resulting representation. This example uses artificial data created by the author.

segmentation of rock, expressed by inner edges (these polygons are referred to as "missing inner edge" polygons in following sections). An inner edge can be one of the following types (see also Figure 4):

– upper inner edge: typically a ridgeline with upper symbols on both sides,
– lower inner edge: typically a valley line with adjacent lower line ends on both sides,
– other inner edge: a general terrain edge where the slope of a cliff changes significantly. This type of inner edge is present in a stepped rock. On one side, its adjacent lower lines end; on the other side, upper symbols are present.

For further processing, a small buffer along the edges is created and is virtually deleted from a polygon. Newly arising edges obtain their type based on edge type: for an upper inner edge, all become upper; for a lower inner edge, all become lower; and for other inner edge, one side becomes lower, the other side becomes upper and the connection between them becomes indefinite. Determining which side is the upper and which is the lower in a certain case, can be arranged by pre-processing the other inner edges to guarantee their unified orientation, for example, with a downward direction on their right side.

After incorporating inner edges, each polygon should be simple or complex. While complex polygons can be split and processed part-by-part, not all simple polygons are suitable for further processing. For this the first requirement is that the length of the upper polyline and the length of the lower polyline are comparable. Contravention of this rule leads to problems when placing lower lines automatically without their mutual intersection. In our implementation, the length of the lower polyline has to be at least half, but no more than twice, the length of the upper polyline. The second condition requires the upper and lower polylines to form most, or at least a half, of the polygon's perimeter. This rule is often broken by elongated polygons following the steepest fall direction, which also causes problems with the placement of lower lines. Simple polygons that do not meet the requirement are excluded from other automatic processing (referred as to "simple inappropriate"), whereas the other ("simple appropriate") will be taken in into account in the following procedures.

5.3 Division of a polygon

For further processing, a delimitation of a strip along the upper polyline is required where the symbols for the upper part will reside. Depending on the size of the symbols to be placed along the upper line, generalization is a reasonable step. Small undulations must be smoothed, using any common generalization algorithm, e.g. Douglas-Peuckner or Bend Simplify. In our experimental implementation, simple line sampling was used. Having generalized the upper polyline, a buffer can be created around it with a size that fits the height of the upper symbols (referred to as the "upper belt"). Upper symbols are placed into the upper belt in the following step. The remaining part of a polygon (referred to as the "lower part") will be decorated with lower lines and lower strokes, that represent the lower part of the rock (see Figure 5). If the polygon was too narrow in comparison with the height of the upper symbols, it would be extended in that place to be at least as broad as the height of the upper symbols.

5.4 Design of symbols for the lower part of a polygon

Lower lines for the lower part of a polygon are prepared in this step. They need to be distributed carefully and densely enough to cover the whole lower part without intersecting each other. In a rather simplified way, lower lines should only consist of a straight line segment, not a general, curved polyline.

As the length of the upper and lower lines can differ and optimal density needs to be achieved, the medial axis (de Berg et al. 2008) between the lower polyline and the border line (between the upper belt and the lower part) was utilised. For narrow polygons or narrow parts of a

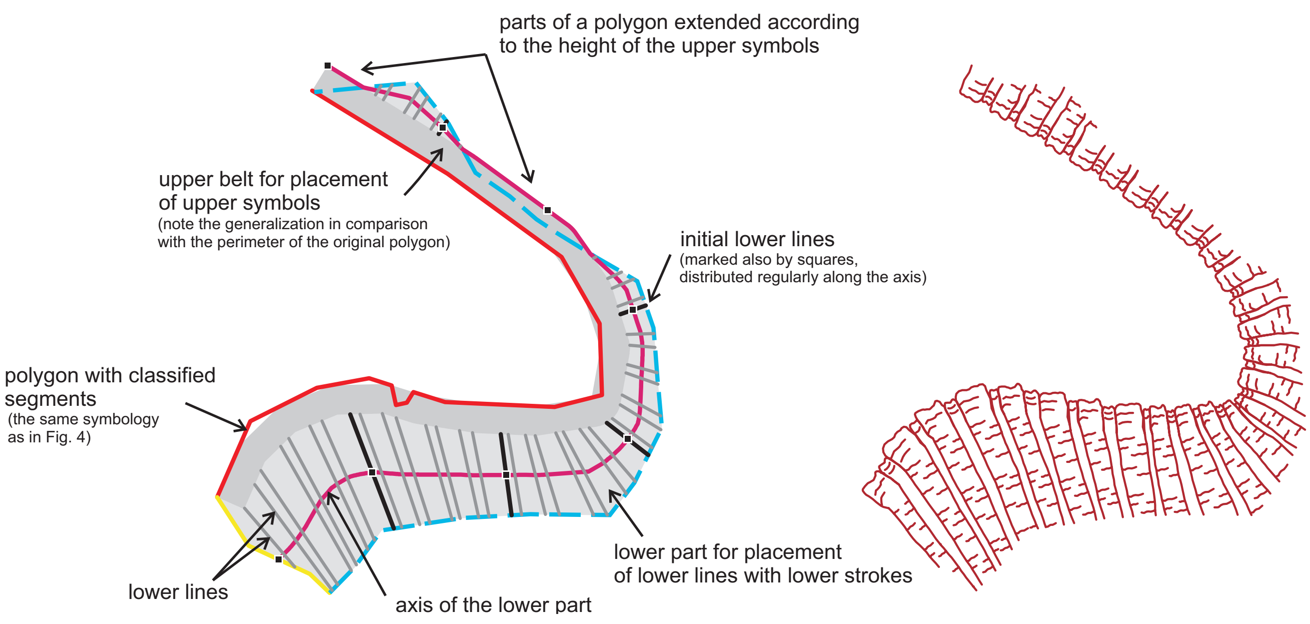
parts of a polygon extended according to the height of the upper symbols
upper belt for placement of upper symbols
(note the generalization in comparison with the perimeter of the original polygon)
initial lower lines
(marked also by squares, distributed regularly along the axis)
polygon with classified segments
(the same symbology as in Fig. 4)
lower part for placement of lower lines with lower strokes
lower lines
axis of the lower part

Fig. 5 Explanation of elements used in single steps of algorithm (left) leading to the resulting representation (right).

polygon, which lack lower lines, the lower limit of an upper belt is used instead. In the first step, initial lines are created which are perpendicular to the medial axis, and far enough away from each other that they do not intersect. Sampling depends on the width of the upper symbol. The sampling distance should be a multiple of the upper symbol's width and possibly the lowest value that prevents the initial lines from intersecting. In complicated shapes, user intervention can be inevitable. In our implementation, a distance of 45 m between the initial lines was used, which is an approximate width of 8 upper symbols from Section 2.

Lines are then added between these initial lines with a simple linear interpolation between their startpoints and endpoints. If the appropriate sampling distance in the previous step was chosen, the lines would not intersect and, at least on the axis, they would achieve the desired density. In our implementation, the space between the initial lines was divided into 8 parts, keeping the ratio 11 : 5 : 9 : 4 : 11 : 5 : 9 : 4, based exactly on the upper symbols described in Section 2. If the value for the initial sampling were too high, linear interpolation would not depict changes to the shape of the polygon and the resulting lines would be improperly oriented.

This simple solution has obvious disadvantages: the required density is reached only in the middle of the polygon, and not necessarily in the upper and lower parts. The lower lines can be over-densified or sparse. This is especially true for extremely curved or twisted polygons. If needed, a user can simply delete parts of the lower lines or draw new parts to achieve a result which is more aesthetically pleasing.

With manual drawing in the Land Survey Office, upper symbols are placed first and lower lines are connected to them. In the proposed algorithm, this process is switched for several reasons. The most important is the fact that the lower part of the polygon can be significantly larger than the upper belt and thus it is more important to keep it graphically consistent. Another argument is that filling the lower part with lower lines regularly would be a harder task if the condition following from the position of previously placed upper symbols is added. Moreover, a positive side effect of this approach lies in the increased irregularity of upper symbols, as intersections of lower lines and border lines (between the upper belt and lower part), corresponding to the widths of upper symbols, are not at exactly regular intervals.

A similar task of filling a polygon with linear features, more or less following the fall direction, is also used for symbolisation of embankments and cuttings with rake-like symbols or slope hachures (cf. Regnauld, Mackaness, Hart 2002).

5.5 Design of symbols for the upper part of a rock

In this phase, upper symbols are put into the area of the upper belt, in such a way that:

a) the sides of the upper symbols continue with lower lines, i.e. the lower end of a symbol will fade into a lower line,
b) the upper symbols are perpendicular to an upper polyline,
c) the upper symbols are not deformed, but alternatively distorted only slightly.

These conditions cannot always be met simultaneously and a compromise is necessary.

To symbolize the upper part of a rock, four symbols were prepared. These symbols were based on the solution described in Section 2. Of course, more or fewer symbols can be used. The exact appearance of a symbol is not important. What is important are

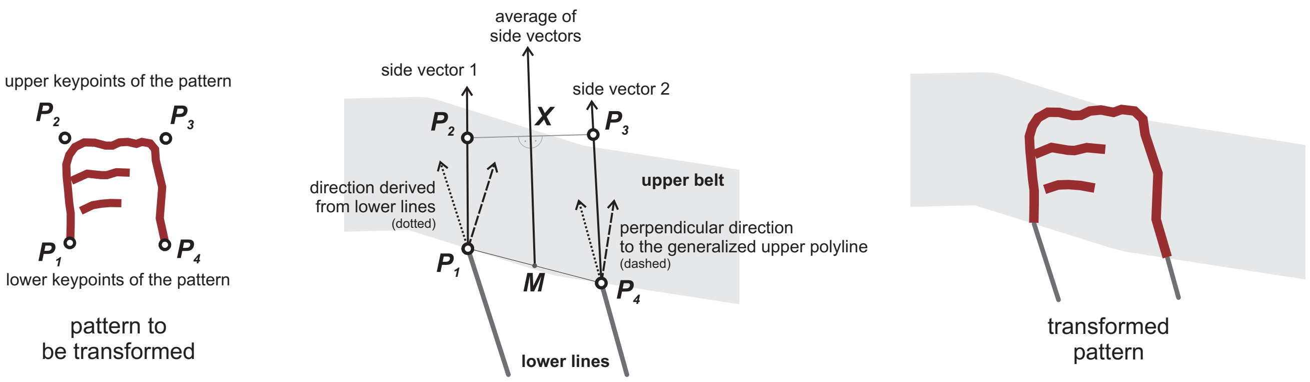
upper keypoints of the pattern
P_2
P_3
P_1
P_4
lower keypoints of the pattern
pattern to be transformed
average of side vectors
side vector 1
side vector 2
X
upper belt
direction derived from lower lines (dotted)
perpendicular direction to the generalized upper polyline (dashed)
M
lower lines
transformed pattern

Fig. 6 Explanation of the placement of upper symbols, which is detailed in the following text.

the keypoints of the pattern, based, for example, on a bounding rectangle of a symbol. Their orientation is also important, i.e. distinguishing between lower keypoints and upper keypoints (see Figure 6).

The placement of each upper symbol into a certain position is based on projective transformation. In the first step, 8 coefficients of this transformation using 4 tuples of identical points are computed, and in the following step the transformation of the pattern is performed.

The lower keypoints of a pattern and the intersection of lower lines with the upper belt, are two pairs of identical points. The other two points, which correspond to the upper keypoints of a pattern, are found using the following concept (see Figure 6). Condition a), mentioned above, would need direction vectors $P_1 P_2$ and $P_3 P_4$ to be the same as for a corresponding lower line; and condition b) needs direction vectors $P_1 P_2$ and $P_3 P_4$ perpendicular to a corresponding upper polyline. The overall orientation of the transformed pattern is an average of these two vectors. This means that the transition between the upper symbol and lower line is not necessarily smooth; however, this is also frequently visible in hand-drawn representations. To meet condition c), a trapezoidal shape of $P_1 P_2 P_3 P_4$ should be avoided in order to keep it rectangular. This is achieved in the following way:

- compute a midpoint M between P_1 and P_4,
- compute coordinates of a point X, which is a shifted M in the direction of the average of side vectors by the height of a pattern,
- compute the intersection of a line, perpendicular to MX, passing through X, with bisectors given by the side vectors.

This means that the bounding box of a transformed pattern can slightly protrude from the upper belt. This is not really a problem as the generalization in step 5.1 alters the extent of the polygon more significantly.

The whole upper belt is filled with transformed patterns, processed successively, i.e. for each tuple of intersections of lower lines with the upper belt, coefficients of transformation are computed. With more patterns, they can be changed regularly (or even randomly). As the distance of lower lines at the intersection of the upper belt slightly differs, every transformed pattern takes a slightly different shape. This is a desirable effect, because irregularity is a desirable feature in this case.

5.6 Dealing with lower strokes

Lower strokes are drawn to help fill the lower part of a polygon. They occupy the space between every odd and even lower line due to their regular pattern and constant length. In the proposed algorithm, a few enhancements are applied:

- variable length of strokes, which helps to fill the area constantly and avoids empty spaces,
- shift of strokes on opposite sides of a tuple of lower lines; this prevents strokes from crossing or overlapping, or from being too dense in a certain part,
- adding more irregularity by a slight random change in the interval at which the strokes are drawn.

Variable stroke length is achieved using a relative value for stroke length, related to the width of the space between the lower lines forming a tuple, with respect to the position of the stroke (as lower lines are typically not parallel). Strokes are drawn perpendicular to the axis. The length of each stroke is computed individually, using the distance d between the startpoint S of the given stroke and the axis. The startpoint S of a stroke is on a lower line; endpoint E of a stroke has the following coordinates:

$$E = S + n \cdot w \cdot d \cdot 2,$$

where w is the relative width of a stroke with respect to the distance of adjacent lower lines ($0 < w < 1$), and n is the normalized normal vector of the axis.

The initial points of strokes can be placed along the lower line regularly, which means that they are placed in a fixed position from a startpoint, and when the line is longer, the positions are "copied" with a given interval. Strokes can also be added randomly, which does not seem to be an ideal solution, as they can overlap with the strokes on the corresponding lower line. A fair compromise is to change the interval slightly for the individual lower line. In our implementation, an interval of 19 m was used (based on the solution from Section 2), altered slightly by a random value following a normal (Gaussian) distribution N(0, 0.5). The relative width and exact position of a stroke within the interval can be found in Table 1; cf. also Figure 7.

Tab. 1 Position and relative width of lower strokes used in the implementation.

lower line on left side		lower line on right side	
position from startpoint (m)	relative width	position from startpoint (m)	relative width
4.3	0.3	1.5	0.6
7.3	0.6	4.5	0.3
14.2	0.5	11.5	0.5

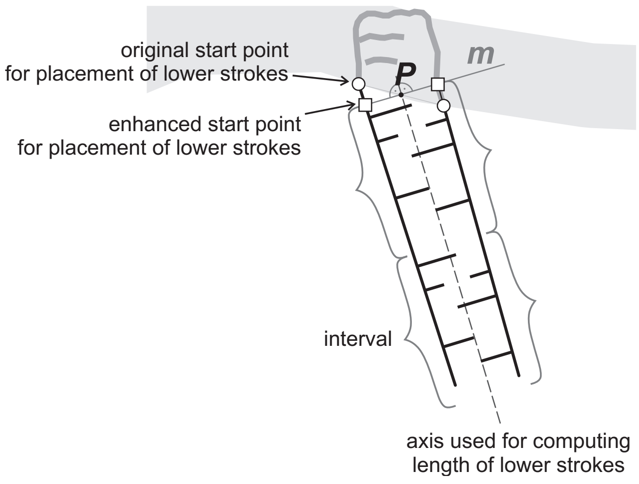

Fig. 7 Enhanced placement of lower strokes, described below in detail.

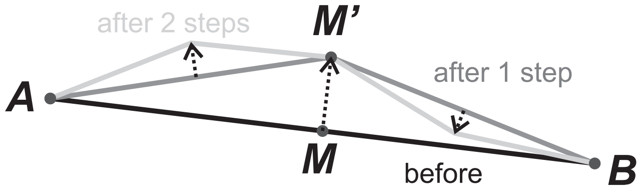

Fig. 8 Random midpoint displacement. The image shows only two steps of the algorithm. The direction and length of displacement is indicated by dotted arrows.

Ensuring that corresponding lower lines are synchronized is favourable in relation to the visual aspect of the result. This means, the strokes do not fill the space in between too irregularly. The greater the difference in length between corresponding lower lines, the more relevant the problem is. This is caused by the difference between the orientation of the upper symbol and the connected lower line, in particular. For this purpose, a virtual start position for placing the initial points of strokes should be changed. For a tuple of corresponding lower lines, the start position for each is determined by the following steps (see Figure 7):

– compute *P* as the midpoint between the startpoints of corresponding lower lines,
– compute line *m* perpendicularly to the axis of the corresponding lower lines, passing through *P*,
– compute the new startpoints for placing the lower strokes as an intersection of *m* with a lower line (possibly elongated upwards).

The described process helps to keep strokes in mutual, visual harmony and fill the lower part consistently.

5.7 Creating a more irregular appearance

Irregular symbols for upper parts are not visually consistent with straight lower lines and lower strokes (cf. Figure 9 top vs. bottom). To increase the aesthetic value of the result, a more irregular appearance is created. While a more irregular appearance is easier and more natural to achieve with hand drawing, in digital processing, the following actions are necessary:

– transforming lower lines using random midpoint displacement,
– fitting lower strokes to transformed lower lines,
– applying patterns to the shifted strokes.

For lower lines, a method called *random midpoint displacement* was utilised. This is often used in computer graphics for procedural modelling (Ebert et al. 2003). It adds a certain amount of "noise" to a drawing. More formally, line segment l is replaced by a sequence of segments p_i, where each p_i has a length less than or equal to a given threshold m. The maximum allowable shift d (the difference between the original segment and the resulting polyline) is also important. For each lower line l (with startpoint A and endpoint B), the following procedure is performed:

– if l is shorter than m, return l as a result
– compute midpoint M of l
– create point M' by shifting M perpendicularly to AB by a random value from -0.5 to $0.5\ d$ (positive value means shift in the right direction from oriented line AB, negative value in the left direction)
– run this procedure recursively for segment AM', with $d = 0.5\ d$
– run this procedure recursively for segment $M'B$, with $d = 0.5\ d$
– merge results of AM' and $M'B$ into a single polyline and return it as a result

The entire process is illustrated in Figure 8 and the result looks more natural than rigidly straight line segments. However, even slight movement of the lower lines causes lower strokes not to be aligned to the lower line exactly. This can be solved by topology-cleaning functions that re-snap endpoints of lower strokes to the altered lower line. This process can be computationally time-consuming.

For the enhancement of lower strokes, a linear conform transformation of a prepared wavy pattern was used. The idea is similar to the one described in Section 5.5, but in this case, only two identical points are used (the startpoint and endpoint of a stroke and the startpoint and endpoint of a pattern), and 4 parameters (shift in x, shift in y, rotation, and change of scale). For each stroke, parameters of transformation were computed and in the following step, transformation was performed on a pattern.

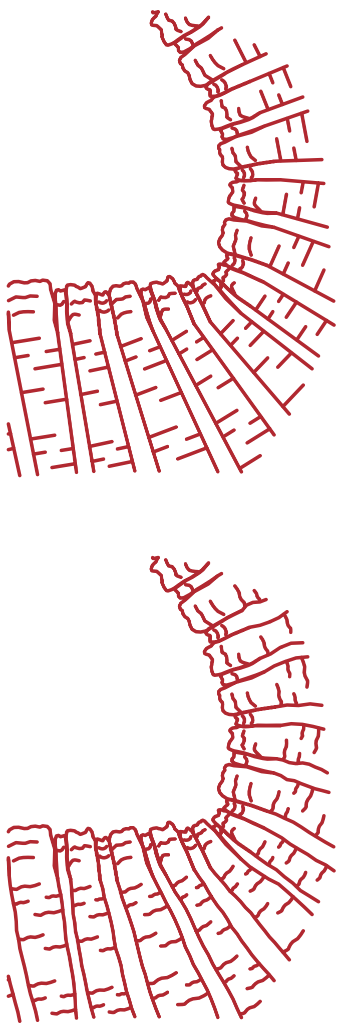
Fig. 9 Adding irregularity of lower lines and lower strokes. Note that some strokes have disappeared.

For lower strokes, transformation of a pattern seems to be a better choice, as the length of the stroke does not vary as much as the length of the lower lines. This is also the reason why the midpoint displacement was adapted for this purpose, as the pattern would have to be too complex for lower lines.

6. Testing and Results

An experimental implementation of the designed algorithm was carried out using the Python and ArcPy module (ArcGIS for Desktop 10.2) and used for testing. It was aimed at creating a representation as similar as possible to the one described in Section 2, i.e. at a scale of 1 : 10,000. Single steps of the designed procedure were performed non-interactively.

For testing purposes, four areas in the Czech Republic that are "rich" in rock were selected. The selection reflected the various types of rocky landscapes in the country. The test sample includes 1,672 objects, covering an area of 3.86 sq. km in total, i.e. approximately 4.2% of the total object count and 3.9% of the total area of all polygons representing rocks in the Czech national topographic database (referred to as ZABAGED. For more details, see ZABAGED, 2014). The test samples were all polygons in one or two sheets of the base map at a scale of 1 : 10,000 (ZM 10). Further details about the test areas can be found in Table 2.

Tab. 2 Description of areas used for testing the algorithm.

test area	total count of polygons	total area of polygons (m^2)	type of landscape
České Švýcarsko	1414	2,518,700	sandstone landscape
Krkonoše	60	293,918	glacially modelled rocks in high mountains
Moravský kras	144	541,874	karren region
Údolí Vltavy	54	502,669	rocks on sides of a deep river valley

Raster DTMs with a cell size of 1 m for test areas were created using interpolation from contour lines in ZABAGED, as the level of detail of both layers is comparable. For this reason, a more detailed DTM from laser scanning data was not used, although it was available. This was done because "old" polygons in ZABAGED are often displaced in relation to the "new" laser scanning data. The main disadvantage of rocky areas with interrupted contours is omitting local elevations or depressions, which cause erroneous results of a preliminary classification of a polygon perimeter.

The main aim of testing was to find out how many polygons will be processed fully automatically and how much and what type of interactive work is required to obtain acceptable results. In the first phase, after fully automatic processing, a thorough inspection of the results was performed manually by comparison with ZM 10. The rock drawing of each polygon was classified as "acceptable", if the result was visually comparable or even better than manual placement by a professional cartographer. The polygon was marked as "requires reclassification" in cases where the result showed errors following the incorrect classification of the perimeter. In this case, for an acceptable result, only a change of attributes (not geometry) is required. Polygons that required some manual editing of intermediate geometry in a certain phase of processing were labelled as "requires editing". Finally, the polygons with very poor results, where neither a change of attributes nor simple editing could help to provide a visually acceptable result, were classified as "unsatisfactory". Overall, the success rate was 57% (949 out of 1,672 processed fully automatically with the acceptable result).

However, there were significant differences between test areas (see Table 3). For the record, the resulting representation consisted of 24,283 upper symbols, 22,482 lower lines, and 42,807 lower strokes.

Tab. 3 Results of automatic processing.

test area	total count of polygons	result of classification (polygon type, see Section 5.2)					result of drawing			
		complex	error	missing inner	simple inappropriate	simple appropriate	acceptable	requires reclassification	requires editing	unsatisfactory
České Švýcarsko	1414	71	47	13	73	1210	838	232	108	32
Krkonoše	60	8	0	0	1	51	28	6	10	7
Moravský kras	144	58	8	0	5	73	62	2	7	2
Údolí Vltavy	54	11	1	0	4	38	21	7	8	2
total	1672	148	56	13	83	1372	949	247	133	43

Tab. 4 Results after manual refinement of the polygon perimeter.

test area	total count of polygons	result of classification (polygon type, see Section 5.2)					result of drawing			
		complex	error	missing inner	simple inappropriate	simple appropriate	acceptable	requires reclassification	requires editing	unsatisfactory
České Švýcarsko	1414	10	0	17	0	1387	1190	0	155	42
Krkonoše	60	8	0	0	0	52	35	0	10	7
Moravský kras	54	2	0	1	0	51	35	0	12	4
Údolí Vltavy	144	9	0	6	0	129	109	0	15	5
total	1672	29	0	24	0	1619	1369	0	192	58

In the second phase of testing, all polygons that were not marked as "simple inappropriate" (and thus remain unprocessed), were examined. Manual reclassification of segments on their perimeters as well as manual correction of segments belonging to the polygons whose rock drawing in the previous phase finished with "requires reclassification", were performed. All simple polygons (even those marked as inappropriate) were forced to be processed. If this manual, but not demanding, work is accepted, the success rate increases to 82% (1,369 of 1,672 processed successfully, see table 4). In most cases, there are only a few segments on a polygon perimeter that need to be altered.

6.1 Discussion

There are several reasons that prevent some polygons from being processed fully automatically. One is the misclassification of the polygon perimeter, often caused by the lack of details in the DTM and in some cases also the generalization used in the interpolation method for finding the steepest direction (the value of *s* described in Section 5.1), which omitted, for example, narrow valleys. Another problem was the excessive level of detail in some polygons (in comparison with a DTM and other map features, especially in the test area Moravský kras), which caused poor classification results. Such polygons need to be generalized before processing.

The fact that the number of really complex (and thus not yet automatically resolvable) polygons decreased to 29 (of 1,672) after manual checking is important. Complex polygons seem thus not to be a serious problem from a global point of view.

Generally speaking, the proposed algorithm works well for elongated features, even of variable width, located on hillsides. For this reason, results from the test area of České Švýcarsko were relatively successful, because that type predominates there. But this cannot be generalized to all sandstone landscapes, as in the case of extremely rugged plateaus the algorithm will fail because this type of landscape requires other means of cartographic representation (cf. Lysák 2015). The bigger and more complicated a polygon becomes, the more problems arise. In some cases, it is very difficult to place straight lower lines automatically, especially if the length of the upper and lower polyline differs or the polygon is too wide. Despite a simple procedure for lower line placement, the intersection of lower lines was not detected. Poorly placed lower lines will cause upper symbols to be deformed, twisted, or impossible to place. This is also caused by bent or crooked upper polylines, where generalization will not be of much help.

To overcome most of the described limitations, the following tasks should be considered more carefully for a more general solution or better results:

– better generalization of a polygon's perimeter classification, which would take more information into account (for example, the inner angles in a polygon),
– the division of a complex polygon into simple parts should be performed automatically,
– a generally better solution for finding initial lower lines, using, for example, inflection points of a medial axis,
– automatic intelligent shortening and/or adding lower lines in areas where the upper and lower polyline lengths differ dramatically instead of excluding them from processing,
– allow lower lines not to be only line segments but general polylines; this allows them to be placed more irregularly in the case of larger polygons. On the other hand, it means complications with intermediate lower lines and lower strokes,
– better handling of the lower lines close to the sides of a polygon, as they do not fill these areas ideally,
– find a generally better solution for narrow rock ridges/stripes, following the fall line direction, as the depiction of those using just one or very few upper symbols and extremely elongated and straight lower lines is very poor.

Moreover, there are also other important issues, not related to algorithm itself, but to its implementation:
– Adding more interactivity would help for practical usability. It is more natural to do the entire process from the classification of the perimeter to the enhancement of lower strokes interactively, feature-by-feature, rather than batch processing, phase-by-phase for all features, followed by ex-post editing. The entire processed area must be checked repeatedly, which is a time-consuming and low-value task.
– The result (polyline representation of upper symbols, lower lines and lower strokes) is very difficult to edit. For efficient editing, intermediate outputs need to be saved and edited. These can also be partly used for the generalization of the result to a smaller scale representation.
– Some parts of the process can be computationally demanding. From these, estimation of the fall direction for tens of thousands of points and locating the medial axis of the lower part are the weakest part of chain.

Further work should aim towards the automation of the ladder manner (as described in Lysák 2015), which shares many features of the solution described above. However, shading may be incorporated so that a plastic and more visually appealing result can be achieved.

7. Conclusion

This article introduced an algorithm for the automation of digital rock drawing, based on a method developed and used in the Czech Land Survey Office. Although the resulting representation is very schematic, it brings the map reader extra information in an illustrative way. Despite it being relatively easy, from a human point of view, for a cartographer to create using simple principles, complete automation of this task is a non-trivial process. Its key steps are thoroughly discussed, indicating that for more sophisticated styles of rock drawing such as in the Swiss manner, the algorithmization of rules will be a difficult and challenging task. The author hopes that this article helps, even marginally, improve the depiction of rocky terrains which are often expressed poorly in contemporary digital maps, and that this article may provide further argument for keeping rock hachures as a useful means of cartographic representation.

Acknowledgements

The author would like to thank all the great people from the Czech Land Survey Office, especially Karel Brázdil, Oldřich Kafka, Danuše Svobodová, and Michal Traurig for their very helpful approach. The work was partially supported by the "PRVOUK P43-Geografie" and "SVV-2010-261 201: Research of physical-geographic changes of natural environment of the Earth" grants.

Author's Note

Experimental scripts used for testing are available upon a request.

REFERENCES

BROWN, L. (2014): Texture Shading: A New Technique for Depicting Terrain Relief. Presentation from ICA Mountain Cartography Workshop, 24 April 2014, Banff, Canada.

DAHINDEN, T., HURNI, L. (2007): Development and quality assessment of analytical rock drawings. Proceedings of the 8th meeting of the ICA Commission on Mountain Cartography, Moscow, Russia.

DE BERG, M., CHEONG, O., VAN KREVELD, M., OVERMARS, M. (2008): Computational Geometry: Algorithms and Applications, Springer-Verlag, Berlin. http://dx.doi.org/10.1007/978-3-540-77974-2

EBERT, D. S., MUSGRAVE, F. K., PEACHEY, D., PERLIN, K., WORLEY, S. (2003): Texturing and Modeling: A Procedural Approach. Morgan Kaufman Publishers 2003.

GEISTHÖVEL, R. (2013): Towards automating Swiss style rock depiction. In: BUCHROITHNER, M. F. (ed.): Proceedings of the 26th International Cartographic Conference (11 pages). Dresden, Germany.

GEISTHÖVEL, R., HURNI, L. (2015): Automatic Rock Depiction via Relief Shading. In: Proceedings of the 27th International Cartographic Conference (8 pages). Rio de Janeiro, Brazil.

GILGEN, J. (2007): Aerial Photos + Photogrammetric Plot = swisstopo's Rock Repre-sentation: Caricatured Mountains? 6th ICA Mountain Cartography Workshop, Mountain Mapping and Visualization. Lenk, Switzerland.

GILGEN, J., JENNY, B. (2010): Digital Rock and Scree Drawing in Vector and Raster Mode. Geographia Technica, Special Issue, 12: 24–31.

GONDOL, L., LE BRIS, A., LECORDIX, F. (2008): Cartography of high mountain areas, testing of a new digital cliff drawing method. In Proc. 6th ICA Mountain Cartography Workshop, pages 71–80, 2008.

HURNI, L., DAHINDEN, T., HUTZLER, E. (2001): Digital Cliff Drawing for Topographic Maps: Traditional Representations by Means of New Technologies. Cartographica, 38, 1 & 2: 55–65. http://dx.doi.org/10.3138/6R12-2GV6-X501-2306

IMHOF, E. (1965): Kartographische Geländedarstellung. DeGruyter, Berlin. http://dx.doi.org/10.1515/9783110843583

JENNY, B., GILGEN, J., GEISTHÖVEL, R., MARSTON, B., HURNI, L. (2014): Design principles for Swiss-style rock drawing. The Cartographic Journal. http://dx.doi.org/10.1179/1743277413Y.0000000052

LYSÁK, J. (2015): Digital rock drawing on Czech topographic maps: current state and the historical circumstances. AUC Geographica, 2015, 2, 193–199. http://dx.doi.org/10.14712/23361980.2015.98

REGNAULD, N., MACKANESS, W. A., HART, G. (2002): Automated relief representation for visualisation of archaeological monuments and other anthropogenic forms. Computers, Environment and Urban Systems, 26: 219–239. http://dx.doi.org/10.1016/S0198-9715(01)00032-1

YANG, N., GUO, Q., SHEN, D. (2009): Automatic Modeling of Cliff Symbol in 3D Topographic Map. Proc. SPIE 7492, International Symposium on Spatial Analysis, Spatial-Temporal Data Modeling, and Data Mining.

YOELI, P. (1985): Topographical Relief Depiction by Hachures with Computer and Plotter. The Cartographic Journal, 22, 2, 111–124. http://dx.doi.org/10.1179/caj.1985.22.2.111

ZABAGED (2014): Fundamental Base of Geographic Data of the Czech Republic [online], [cit. 2014-10-31]. Available from: http://geoportal.cuzk.cz/Default.aspx?mode=TextMeta&text=dSady_zabaged& side=zabaged&menu=24

RESUMÉ

Algoritmus pro automatizované znázornění skal ve stylu používaném na českých topografických mapách

Článek podrobně popisuje možnosti automatizace metody pro znázorňování skal s využitím prostředků digitální kartografie, vyvinuté Zeměměřickým úřadem a používané na topografických mapách jím vydávaných. Podstatou této metody je vyplňování půdorysu skalního útvaru liniemi, které mají připomínat stylizované skalní šrafy. Tento postup je dosud prováděn ručně a je poměrně časově náročný. Na základě analýzy práce kartografů a výsledné reprezentace na mapách je proto navržen a detailně popsán algoritmus, který tuto činnost umožní do značné míry zautomatizovat. Vlastnímu návrhu algoritmu předchází srovnání s podobným typem prací, které se ale pokouší více či méně úspěšně automatizovat jiné styly skalní kresby. V článku jsou popsány vstupy a výstupy algoritmu, je provedena podrobná analýza klíčových fází zpracování, ilustrovaná na konkrétních případech. Vytvořený algoritmus byl implementován s využitím ArcGIS 10.2 a úspěšně otestován nad reálnými daty. Součástí článku je též diskuse dosažených výsledků včetně možných vylepšení.

Jakub Lysák
Charles University in Prague, Faculty of Science
Department of Applied Geoinformatics and Cartography
Albertov 6, 128 43 Praha 2
Czech Republic
E-mail: lysak@natur.cuni.cz