



Approximate local reflection symmetry of projected land cadastre data

Ivana Kolingerová¹ · Ondřej Anděl¹ · Eliška Mourycová¹ · Pavel Slavík² · Lukáš Hruďa¹ · David Podgorelec³ · Borut Žalik³ · Ivo Malý² · Martin Maňák¹

Received: 1 December 2023 / Accepted: 14 November 2025 / Published online: 4 December 2025
© The Author(s) 2025

Abstract

Symmetry is an essential feature of many geometric objects. However, the world also contains many asymmetrical or approximately symmetrical objects. Detecting approximate symmetries is a rather weakly defined problem, as computer-detected approximate symmetry may not correspond to human opinion. The situation is even worse if the symmetry is not global but local. This paper investigates whether approximate local reflection symmetries found by a computer in real data are acceptable for human observers. To answer this question, a new simple approximate local reflection symmetry detection is proposed and run on land cadastre data in the form of planar point sets. The resulting symmetries are subject to user tests to study human acceptance of approximate local symmetry. The results show a relatively good correlation between symmetry detected by computers and perceived by humans. This finding provides a solid foundation for integrating both approaches in specific applications. To achieve this, further research is needed on how to utilize specific aspects of human symmetry perception in computer solutions, so that computer symmetry detection can better approximate human perception.

Keywords Symmetry · Geodata · Computer graphics · Human perception

✉ Ivana Kolingerová
kolinger@kiv.zcu.cz

¹ Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia in Pilsen, Pilsen, Czech Republic

² Department of Computer Graphics and Interaction, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic

³ Faculty of Electrical Engineering and Computer Science, University of Maribor, Maribor, Slovenia

1 Introduction

According to a formal geometric definition, an object has symmetry if a geometric transformation (such as translation, scaling, rotation, or reflection) maps it into itself. The possible types of symmetries depend on the set of geometric transformations available and what object properties should remain unchanged after the transformations are performed – see Martin (1996).

Symmetry can be recognized in many human-made objects as well as in nature. According to the American scientist Lightman (2014), human brains strive to see things symmetrically. “The reason must be psychological,” he says. “Symmetry represents order, and we crave order in the strange universe we find ourselves in ... [It] helps us make sense of the world around us. Symmetry perception is hard-wired in the human brain. That means we register the presence of symmetrical features in our field of vision before we realize what we are looking at. The presence of symmetry helps us to understand the perceived scene and analyse it more efficiently”.

The most common type of symmetry is reflection symmetry, produced by the reflection transformation. Often, an object is not globally symmetrical but exhibits local symmetry (i.e., only part of the object is symmetrical). Although more difficult to find, it may still be helpful both for human perception and applications (see example in Fig. 1a).

Many methods have been developed to automatically find symmetry in geometric models for global and reflection symmetry. The methods were typically designed for human-produced symmetrical objects. Not much has been done for geodata that are

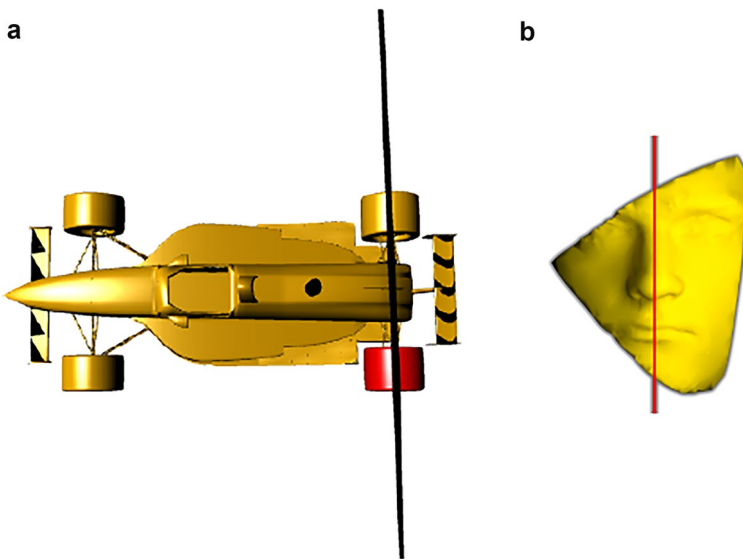


Fig. 1 Examples of symmetry, **a** the plane of local reflection symmetry (black) for one wheel (the red one) - Uhrová (2021), the model with courtesy to Zhou and Jacobson (2016), **b** the plane of approximate global reflection symmetry (red) – Hruđa (2021), the model with courtesy to Chalás et al. (2014)

of natural origin, although information about their symmetry could eventually help their understanding or efficient compression.

Symmetry detection by humans has been the subject of research for decades. Many factors influence this perception. Fundamental works in this area are Zabrodsky (1990), Zabrodsky and Algom (1994), Sundaram et al. (2022). The complexity of human symmetry detection in specific applications has been investigated for many years, with the results that both global and local symmetries are essential, e.g., Palmer (1985).

Besides being an essential concept for human perception, symmetry has important applications in science and engineering. For such applications, it is often necessary to expect and use approximate symmetry because the model may have additional or missing data ‘spoiling’ its symmetry, see example in Fig. 1b). Even on a shape with no missing or additional data, the sampled points only exceptionally contain pairs of mutually symmetrical points because their sampling did not respect symmetry.

The research in symmetries has numerous applications in science and engineering. Therefore, it is important to work not only with artificial models but also with application data. In the research described in this paper, we utilize the data from geoinformatics.

The main research question addressed in this paper is whether approximate local reflection symmetries in geodata found by a computer are acceptable for human observers. The question is addressed by two entangled tools: First, a suitable approximate-local-symmetry method is proposed and applied to real geodata. Second, the results are evaluated through user tests, allowing for the study of human acceptance of approximate local symmetry in real data. The novelty of the approach used to investigate the main research question is combining a novel, albeit simple, local reflection symmetry detection method, human evaluation, and the use of geodata. This combination is, as far as we know, unique.

The data used in experiments were sets of 2D (or projected 3D) points corresponding to a terrain model. The point sets typically contain some shapes corresponding to landscape features and human-made constructs, but the main impact on the observer is due to the outline of the shape of this point set. Therefore, it makes sense to concentrate mainly on the outer shape. As the terrain point sets typically have only approximate symmetry, it is difficult to quantify the quality of the results formally, as there is no ground truth, so user evaluation is more reasonable than using some measure.

The proposed solution for the local symmetry of the described data is based on the method of global symmetry by Hruda et al. (2022a), developed mainly for 3D surface models described by point clouds but usable also for points sitting in a plane embedded in 3D, and well-tested. Two modifications of the method are presented, which can find vertical planes of approximate local symmetry in terrain data. The modifications consist in uncomplicated identification of the concavities in the shape of the data domain and using the weights in the original method to find planes producing approximate local symmetry. The method suits data sets with a rather complex, usually concave, polygonal domain shape. For such a shape, it makes sense to concentrate on some symmetry of the domain shape (outline); if the shapes were simple, such as a rectangle, inspection of its symmetry would bring nothing interesting. The method is not tightly bound to geodata, so it can also be used in another context for

lossy compression or reconstruction, shape matching, or as a tool to understand the data better.

The paper is organized as follows. Section 2 brings background, namely, existing symmetry detection methods, a more detailed description of the method of global symmetry by Hrudá et al. (2022a), and a survey of symmetry applications. Section 3 describes the proposed method in two modifications. Section 4 presents the results, and Sect. 5 concludes the paper.

2 Background

2.1 Symmetry detection

From the possible symmetries, global reflection symmetry is most often needed; therefore, many methods to detect it came into existence. A survey can be found in Mitra (2013). It is impossible to say which of the methods is best; as usual, mathematically and algorithmically more complex approaches bring higher efficiency and vice versa. The most frequently cited method of global reflection symmetry in 3D is by Sipiran et al. (2014). This method processes (possibly incomplete) triangle meshes. It detects local features of the input object based on the maxima of the function for heat diffusion. Pairs of those feature points with similar values of the heat diffusion function generate candidates for the planes of reflection symmetry. Other points then vote for the candidates, and from the candidates, the plane with the highest satisfaction of points is taken as the most proper plane of reflection symmetry.

Podolak et al. (2006) use a planar reflective symmetry transform (PRST) that captures a continuous measure of the reflection symmetry of a shape concerning all possible planes. Then an iterative refinement algorithm finds the local maxima of the transform based on which the centre of symmetry and the best plane are obtained.

Li et al. (2016) developed a view-based reflection symmetry detection method based on the viewpoint entropy features of a set of sample views of a 3D model. The viewpoints are sampled on a sphere around an input model. For the views from these viewpoints, entropy is computed, resulting in an entropy distribution sphere used for symmetry plane detection.

Schiebener et al. (2016) use symmetry to complete partially known shapes for robotics applications. The method works for point clouds and also needs to know a point cloud in the neighbourhood of the object. It supposes that a 3D object usually stands on some supporting plane, which helps to restrict the space for symmetry planes search. Gaps along the sides in the direction of the view axis are closed by linear interpolation. Other global reflection symmetry methods, mostly geometry-based, can be found, e.g., in the papers by Sun and Sherrah (1997), Karakala et al. (2019), and Korman et al. (2015).

It is also possible to employ artificial intelligence methods, namely neural networks, e.g., the method by Ji and Liu (2019), Gao et al. (2021), and Shi et al. (2020). These methods provide the usual set of advantages and disadvantages of neural networks: application is fast in case one has an already trained network, but its training needs time and a huge amount of data. A completely different object may also ‘spoil’

the neural network while its success on such data is uncertain. A new promising approach in neural networks may be to train them so that they may “consider” human experience in symmetry detection, see Krippendorf and Syvaeri (2020).

Local symmetry detection is a more complex problem as evaluating which candidate reflection planes are essential and which are ‘too local’ (such as the planes of symmetry of a button on a jacket) is demanding. The methods often require a mesh as an input - Ecins et al. (2017), are intended for geometric models - Mitra et al. (2006), or seem to be relatively slow - Lipman et al. (2010). The Podgorelec et al. (2023) method suits geodata, but the raster-based algorithm has limitations. The method by Žalik et al. (2023) produces a geometric shape characterization, which is a generalized skeleton not equal to the axis of symmetry. Hruda et al. (2023) combine the PCA descriptor point matching with the density peak location algorithm; the method is mainly intended for a scene of geometric objects. Therefore, none of the methods suit huge input point sets, typical terrain data.

Simari et al. (2006) present an algorithm detecting local reflection symmetries of 3D triangle meshes. Three orthogonal planes going through the centroid are found, from which the one best reflecting the points is kept, and a support region is defined with strong enough symmetry according to this plane. This region is left out, and the method is used iteratively; in this way, local symmetry is produced.

The ground for the solution proposed in this paper is our method developed recently that finds the best-fitting plane of global symmetry for the set of points in 3D. More details about the method and its efficient realization can be found in Hruda et al. (2022a). The software implementing the method can be downloaded from Hruda et al. (2022b).

The method needs neither a mesh nor other auxiliary geometrical information, such as normal vectors (although such extra information could be utilized). It can process huge data sets and produce more planes of reflection symmetry, sorted decreasingly according to the symmetry measure.

The task is solved by maximizing an objective function called symmetry measure, defined for a set of points $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, $\mathbf{x}_i \in E^3$ as

$$s_X(\mathbf{T}) = \sum_{i=1}^n \sum_{j=1}^n w_{ij} \phi(\|\mathbf{T}(\mathbf{x}_i) - \mathbf{x}_j\|) \quad (1)$$

where \mathbf{T} is a geometric transformation (a reflection in our case), $\phi(\cdot)$ is a similarity function, providing information on how close the point \mathbf{x}_i , transformed by the transformation $\mathbf{T}(\cdot)$, is to the point \mathbf{x}_j . The weights w_{ij} of point pairs $\mathbf{x}_i, \mathbf{x}_j$ can be ignored and set to 1, but their use provides more flexibility in the method. These weights are essential for our purpose as they allow for more influence on some points.

Searching the plane of symmetry is understood as an optimization problem – the plane best satisfying the input set of points is chosen; such an approach is general enough to incorporate various understandings of symmetry, using different symmetry functions and weights dependent on the type of data investigated. Primarily, it was tuned with the prevailing applications in mind, such as incomplete symmetrical model reconstruction (e.g., in the case that some errors in the data are to be recognized and repaired) or compression (e.g., for the so-called generalization when the

data are to be reduced to a smaller scale). Thus, missing or imprecise pairs of points decrease evaluation of the corresponding symmetry plane but enable us to find the best solution even for approximately symmetrical objects, which, however, might be refused by a human observer as non-symmetrical due to the user's previous experience. This aspect is significant for the data addressed in this paper.

2.2 Symmetry applications

The research on symmetries has significant application potential across diverse fields, contributing to scientific discovery, technological innovation, artistic expression, and societal advancement. Exploring the principles of symmetry and their implications brings new insights and solutions to complex problems.

Focusing on applications in science and engineering, we can mention geometric data compression, e.g., Simari et al. (2006), symmetrical editing, e.g., Martinet et al. (2006), and model reconstruction, see Mavridis et al. (2015), Schiebener et al. (2016), where on the assumption of symmetry, the whole object model can be recovered from partial data, which possess a certain degree of symmetry. Symmetry analysis is used in signal processing, see, e.g., Girault (2024). Symmetry-based techniques help uncover relationships in large-scale data sets, see, e.g., Murtagh (2008), enhancing data visualization, clustering, such as Bandyopadhyay and Saha (2013), and anomaly detection, see Qin et al. (2020). Namely, local symmetries can be used for pattern recognition – Bigün (1988) or image matching using local features - Hauage and Snavelly (2012), Leng et al. (2019), face detection – Sun et al. (1998).

Symmetry can also be used as a tool for image compression. Many applications exist in medicine and biology, where this compression type has found its place. For example, the paper by Bairagi (2015) describes a method that allows efficient compression in medical imaging where huge amounts of data are to be processed. This method is based on the detection of the central symmetry axis in the image (in this case, the vertical one). With such a symmetry axis, we can store (or transmit) only half of the image (as the other half is symmetric). If a lossless compression is needed (often in medicine), we can create an additional file where differences between both halves are stored.

In the applications mentioned earlier, we work both with global symmetry and local symmetries that help us detect similar parts of the model that may contribute to the compression strategy used for this model. Approximate symmetries may also play their roles. If we identify several objects with approximate symmetries, we may declare them as perfect ones. It allows us to consider them in the data compression process, which will result in higher efficiency of the compression performed.

Applications of local symmetry in geodata are for typical tasks, such as segmentation and object recognition. The semantic segmentation assigns semantic labels (classes), such as ground, buildings, or vegetation, to the groups of points or pixels. Applications such as city planning, precision agriculture, road monitoring, or autonomous driving certainly require the highest possible accuracy; thus, every new improvement counts. The developed solutions can still profit strongly from detected symmetries in the data. Symmetry is frequently used in 3D architecture, street or

network of streets reconstruction is undoubtedly the most common application of detected symmetries.

Clode et al. (2007) represented one of the earliest approaches for detecting roads from LiDAR data, using both the intensity and range data. The symmetry is implicitly detected and employed in centerline, width, and direction calculations. The method additionally provides vectorization. Gézero and Antunes (2019) recently presented an automated 3D extraction of linear railway elements from mobile LiDAR point clouds based on the symmetry of the standard railway construction profile. Kootstra et al. (2009) use local symmetry for landmark selection: according to their research, interest points selected using symmetry are more robust to noise and contrast manipulations, have slightly better repeatability, and result in better overall SLAM (Simultaneous Localization And Mapping) method performance. Wang et al. (2015) proposed an automatic decomposition and modeling of the compound buildings from LiDAR data and aerial imagery into semantic primitives with fixed parametric forms by exploiting local symmetry contained in the building structure. Tu et al. (2017) utilized monitoring changes in the symmetry of windows on a segmented building facade in disaster assessment and management.

Applications can be seen in architecture and urban planning, see Mehafy (2020), as symmetry has an eye-pleasing effect. As for the cadaster data, we have to stress that this type of data is used in urban planning, and the detection of symmetries by humans may be a cumbersome task. Computer detection can simplify this task considerably as it may provide humans with hints on where to find symmetry. Kerber et al. (2013) presented a method for detecting local symmetries in huge point clouds of 3D city scans. The key idea is to design a feature space in which nearby points have symmetric geometry with high likelihood and then solve the nearest-neighbour problem in a lower dimension. The method robustly handles noisy real-world scanner data, data from multiple images, or a combination.

As geodata (e.g., cadastre data) are typically huge, compression based on symmetry can find its place in this area. Haurert (2012) describes symmetry in city maps (based on cadaster data) for map generalization. Cartographic generalization, or map generalization, see, e.g., Jiang et al. (2011), includes all changes made when one derives a smaller-scale map from a larger-scale map or map data. It is a core part of cartographic design. The traditional approach requires a person to select and simplify the level of detail on their map to optimize how information is presented. As the symmetry detected allows the discovery of the data structure, including repetitive patterns, it may serve as a tool for consequent data compression (as stated above). Working with approximate symmetry, we obtain an essential tool that allows us to consider approximately symmetrical objects as symmetrical (because the map resulting from the map generalization process does not contain tiny details (as was the case in the original map on a larger scale).

We cannot expect perfect symmetries when working with application data, such as geodata. The geodata are very challenging because their captured symmetric parts are often obscured, shadowed, or detected from unpredictable directions only after several reflections, resulting in partial and noisy data that calls for several adaptations of the symmetry detection methods. To be able to utilize symmetry in geodata, we must be able to provide local, approximate symmetries for incomplete data.

3 The proposed method

The ground method for the proposed solution was developed to find planes of global symmetry of a geometrical object represented by a set of points sampled from the surface in 3D or from the object outline in 2D; the method was already shown to be fast and reliable, see Hruđa et al. (2022a).

For local symmetry, we added the detection of those parts of the model that “stick out,” so they are candidates for some perceivable local symmetry. We will call such a part of the model a peninsula in the following text, see Fig. 2. Which peninsula is essential to deserve processing strongly depends on the context and the observer, and there is no ground truth. We developed two peninsula-detection algorithms for this task; the first is based on the K-means clustering, and the second utilizes the object’s axis-aligned bounding box (AABB). The clustering was chosen to make groups of mutually close points. As there are no special requirements for the resulting clusters, which are just an auxiliary tool here, any clustering algorithm could be used; for simplicity, we chose the widely-known K-means algorithm. The AABB structure was selected as a simple tool to find points with extremal coordinates or lying near the extremes. These points are suitable candidates to start the breadth-first search of peninsulas. Both algorithms will be described for planar points, but they can also be used for 3D data in case of a different input. The algorithms are described in Sect. 3.1 and 3.2.

When peninsulas have been detected, the search of symmetry planes on the input model can be applied, as described in Sect. 2, with the weights of points in this part



Fig. 2 An example of a peninsula (red) with an axis of approximate local symmetry (black)

of input data increased to motivate the symmetry detection algorithm to prefer local symmetry in this part. A proper choice of weights in Eq. 1 will be addressed in Sect. 4.

3.1 K-means-based peninsula detection

The algorithm works as follows. First, the input points are clustered; various clustering algorithms could be used. Usually, higher flexibility (e.g., more variety in the shape of clusters or no need to set the number of clusters in advance) are paid by higher complexity. We decided on K-means due to its well-known properties and simplicity. More sophisticated solutions can be found, e.g., in Žalik and Žalik (2009) and Bayer et al. (2023).

The found clusters are evaluated on how much they correspond to peninsulas, i.e., whether they correspond to the parts that are narrow and far from the point of gravity. The best-evaluated clusters are kept for further processing. See more details in Algorithms 1.

Algorithm 1 K-means-based peninsula detection

- Input: A set of points \mathbf{x}_i , $\{\mathbf{x}_i = (x_i, y_i)\}$, $i=0, \dots, n-1$, k - the desired number of clusters to be created, l - the number of best clusters to be kept.
 - Output: l best-evaluated clusters of points corresponding to l peninsula.
1. Initially, k random points from the input set are selected as cluster centres.
 2. For each point, the closest cluster centre is found, and the point is assigned to this cluster. The cluster centre is recomputed as the cluster centroid.
 3. It is checked if any of the points should change their assignment to clusters. If so, the computation goes to step 2.
 4. Each cluster's quality is evaluated by calling *Cluster quality evaluation* function (see below).
 5. l clusters with the best evaluations are returned.

Function *Cluster quality evaluation*.

- Input: A cluster containing n_c points, its centroid \mathbf{x}_c
 - Output: cluster evaluation *eval* (a lower value is better).
1. The point of cluster \mathbf{x}_f , furthest from the centroid \mathbf{x}_c , is found.
 2. A line L given by \mathbf{x}_f and \mathbf{x}_c is constructed.
 3. All cluster points are orthogonally projected onto L , and the sum s of the distances between the points and their projections are computed.
 4. $eval = s / n_c$.

See Fig. 3 for an illustration of cluster quality evaluation. The points of a cluster to be evaluated are visible in Fig. 3a). In Fig. 3b), the centroid \mathbf{x}_c is computed (the black point). In Fig. 3c), the point \mathbf{x}_f furthest from the centroid is found (the red point). The line L given by \mathbf{x}_c and \mathbf{x}_f is constructed and shown in Fig. 3d). Points in the cluster are

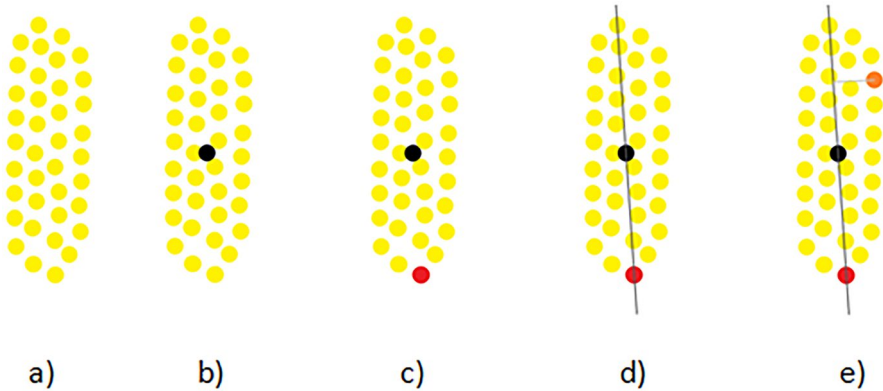


Fig. 3 Illustration of cluster quality evaluation; points are represented with small circles; **a** input points (yellow), **b** the centroid (black) was computed, **c** the point furthest from the centroid (red) was computed, **d** a line given by these two points was computed (black), **e** an example of a point (orange) and its projection

projected onto the line, and their distances are used for the evaluation (an example of a point being projected is coloured orange in Fig. 3e).

3.2 Axis-aligned bounding box-based (AABB) peninsula detection

The algorithm is based on the idea that the peninsula's extreme points are probably far from the centre of gravity of the input data; maybe they even touch the bounding box of the input point set. Although this supposition is neither necessary nor satisfactory condition and is not rotation-independent, it may help detect parts of the data sets suitable for the local symmetry inspection. See Algorithm 2 for more details.

Algorithm 2 AABB algorithm

- Input: A set of points \mathbf{x}_i , $\{\mathbf{x}_i=(x_i, y_i)\}$, $i=0, \dots, n-1$; c – user parameter, setting the relative distance to which points should be considered as one segment, usually $c=1$; $MaxSegSize$ – the maximum allowed size of a segment, $dist(\mathbf{x}_i, \mathbf{x}_j)$ – a Euclidean distance of two points \mathbf{x}_i and \mathbf{x}_j
 - Output: l segments S_i , $i=0, \dots, l-1$ corresponding to the best found peninsulas.
1. The bounding box and the centre of gravity \mathbf{c}_g of the input points are computed.
 2. $distDelta=0.1$ *the length of the shortest side of the bounding box.
 3. The set XB of those points \mathbf{x}_i , which are incident with the bounding box, is computed.
 4. For each point \mathbf{x}_i : add all points \mathbf{x}_j which fulfill the condition $dist(\mathbf{x}_i, \mathbf{x}_j) < distDelta$ into an auxiliary table AT (the table will be used to accelerate the processing of neighbourhood of the point \mathbf{x}_i).
 5. Points \mathbf{x}_i , $i=0, \dots, k$ in XB are sorted in descending order according to $dist(\mathbf{c}_g, \mathbf{x}_i)$.
 6. $avgDistToCg$ is computed as the average distance of the \mathbf{x}_i points from \mathbf{c}_g .

7. BFS traversal is started from each $\mathbf{x}\mathbf{b}_i$ in XB using the information in AT , and visited vertices are added to S_i ; the traversal is stopped if the next point to be seen (\mathbf{p}) satisfies $\text{dist}(\mathbf{c}_g, \mathbf{p}) < c * \text{avgDistToCg}$.
8. //An alternative stop condition for step 7 is: BFS traversal is stopped if the next point to be seen causes $|S_i| \geq \text{MaxSegSize}$.
9. All segments S_i are evaluated using Algorithm 2, and l of them are considered the output peninsulas, where $l \leq k$ and l is user-defined.

See Fig. 4 for an illustration of how the AABB algorithm works. In Fig. 4a), an example of an input point cloud is shown. In Fig. 4b), the AABB of the input point cloud is constructed and represented by a grey box around the input points. In Fig. 4c), the set XB of points incident with the AABB is found (green). A starting point is selected from the set XB – in this case, it is the downmost point (red, Fig. 4d). BFS traversal is started from this point and stopped after one of the stop conditions is met (Fig. 4e-g).

4 Experiments and results

Both algorithms were implemented in C++ and tested on a computer with a dual-core processor, Intel CoreTM i5-7200U CPU with frequency 2.5 GHz, 8GB RAM, and Windows 10. The testing (cadastre) data are shown in Fig. 5. All these data sets have a non-convex outer shape and are only approximately symmetrical, if at all. As these data sets are 2D projections of 3D data, their plane of symmetry degenerates to a 2D

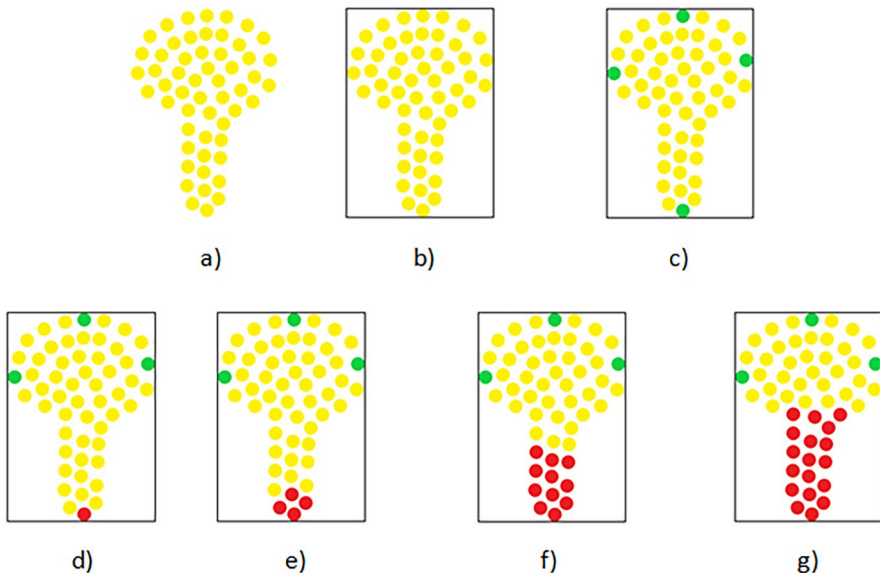


Fig. 4 Illustration of the AABB algorithm at work; points are represented with small circles; **a** input points (yellow), **b** the AABB was computed (grey), **c** points incident with the AABB were found (green), **d** a starting point was chosen from the green points (red), **e, f** two phases of the BFS traversal, **g** the BFS traversal result

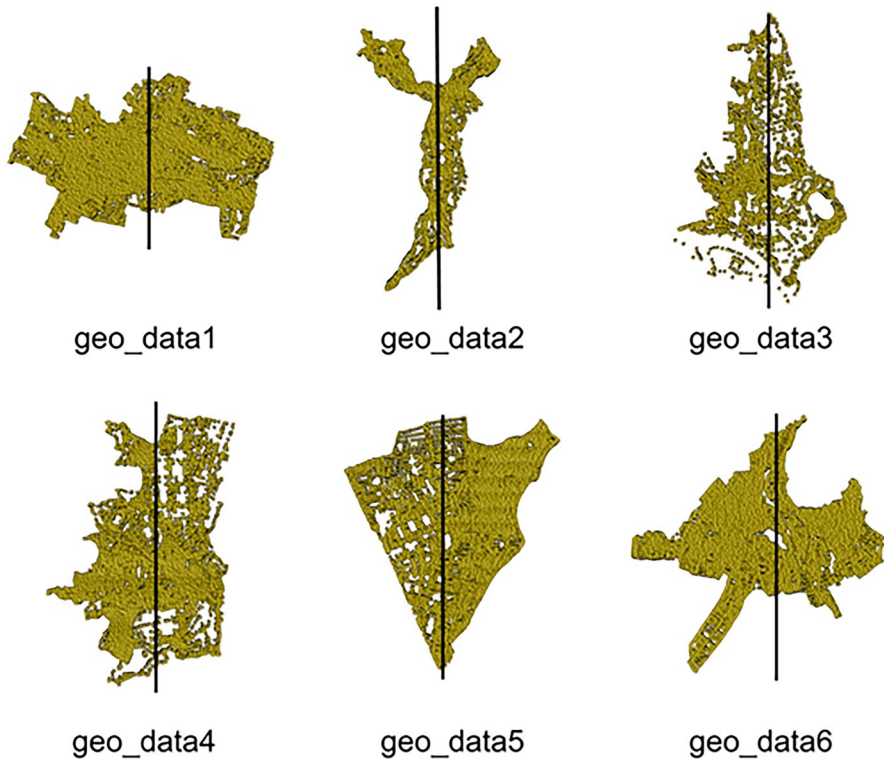


Fig. 5 The tested data sets (points represented by yellow balls; the thick black line segments are the axes of approximate global reflection symmetry)

axis. In Fig. 5, they are depicted together with the axis of approximate global symmetry as detected by the ground method of global symmetry; the orientation of the datasets was chosen to have this axis vertical. Typical behaviour of the approximate symmetry detection is visible here: the method tries to minimize the number of points “unsatisfied” with the axis; missing points are not an obstacle as the data are allowed to be incomplete. Most of the results are agreeable for the observer; the worst result is the `geo_data6` data set, where a manual choice would prefer an axis such as in Fig. 2.

The outer shapes of these data sets contain various non-convexities where approximate local symmetry can be found. For both algorithms, we tested two weight settings. The former sets the weights of points outside the peninsula to zero and for the peninsula points to one. This choice turns off the other parts of the data than the particular peninsula. The latter setting is to consider the out-peninsula points with weights one while the peninsula points have higher weights, so they have a more substantial influence. We put this value to five based on experiments, as this value provided the best results in preliminary experiments. The possibility and necessity to set weights can be considered either an advantage or disadvantage of the proposed method: on one side, the weights enable to put more or less stress on the influence of the whole or the local part; on the other side, some experiments with the particular data may be needed to set the parameters properly.

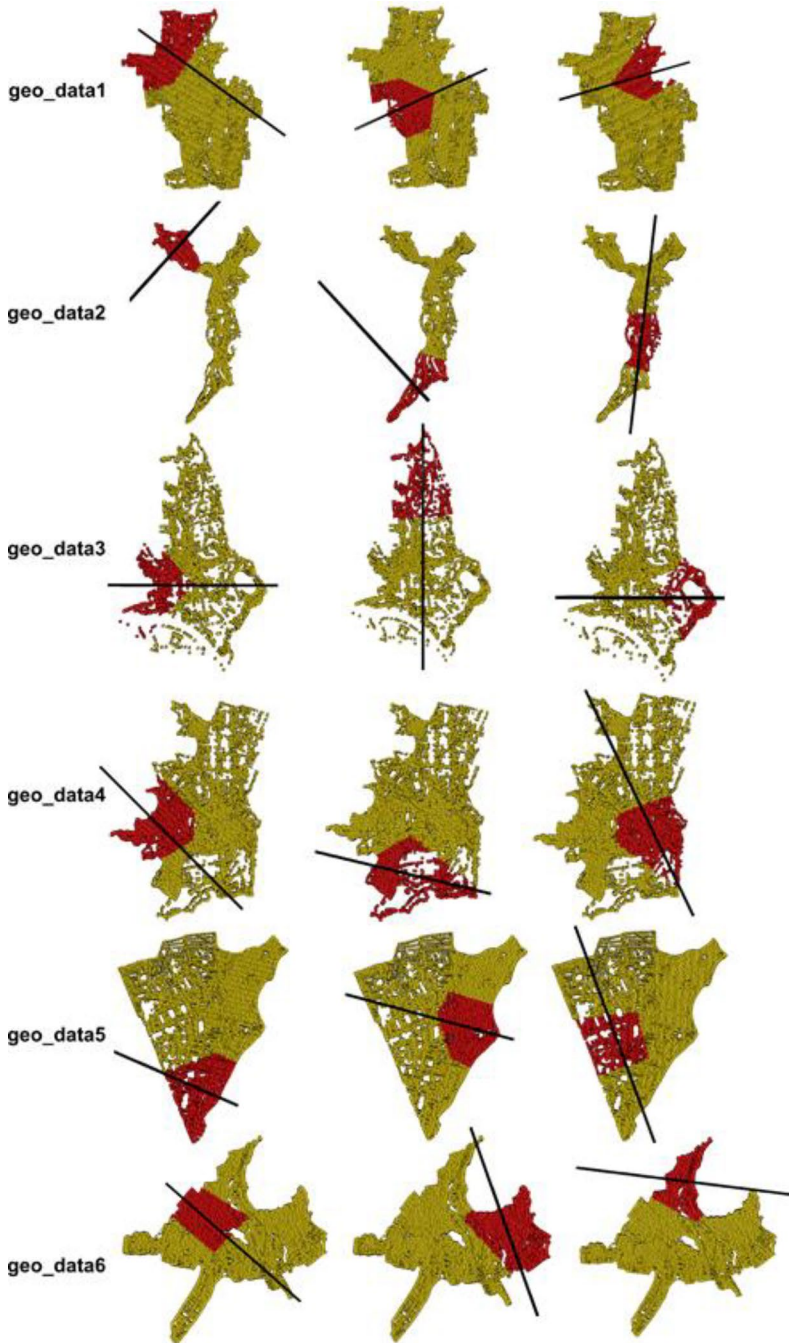
The results for the K-means-based method are shown in Fig. 6 and for the AABB method in Fig. 7, respectively. The results show that both algorithms can detect the essential parts of the outer shape that stick out (peninsula), on which local symmetry can be inspected using the method of reflection symmetry search. When we look at the results in both figures, it is difficult to say which algorithm works better. Formal measuring the quality of achieved results is difficult as there is no ground truth. As the symmetry is only approximate and the ground method can ‘imagine’ eventual missing points, the methods can provide more distinctive local symmetries for different settings of weights. For example, comparing Fig. 6b) the second row with Fig. 6a) the second row (“geo_data2”) clearly illustrates the differences between both weight settings: in the first case, only the peninsula was considered, while in the second case, the whole shape was considered. However, not all symmetries found by a computer correspond to what the user would choose based on perception. To test this difference in computer and human evaluation, we processed the results by user tests.

The user tests were done by 30 anonymous users (27 men and three women; 20 university students and 10 employees, aged 23 to 59; the users had no visual defects not compensated by glasses). The participants were volunteers and never participated in experiments of this kind. All of them were computer scientists with no geodetic background. However, most of them work in computer graphics, so they are used to produce and perceive various visualizations.

We prepared an application for the experiment where images from both methods’ results on various data sets are mixed. In one moment, the users have four screenshots on the screen and are to mark (evaluate) each image according to their degree of symmetry by an integer in the scale 1.5 (1 is better). Displayed images were grouped into pages based on their original data and the similarity between the peninsulas detected. For every user, the order of pictures on one page is randomized. The reason is to minimize the influence of their location in the evaluation process. All images were rotated to have the detected axis vertical as people are more sensitive to vertical symmetry, and we wanted to eliminate this potential bias in the answers. Figure 8 shows a screenshot from the testing application.

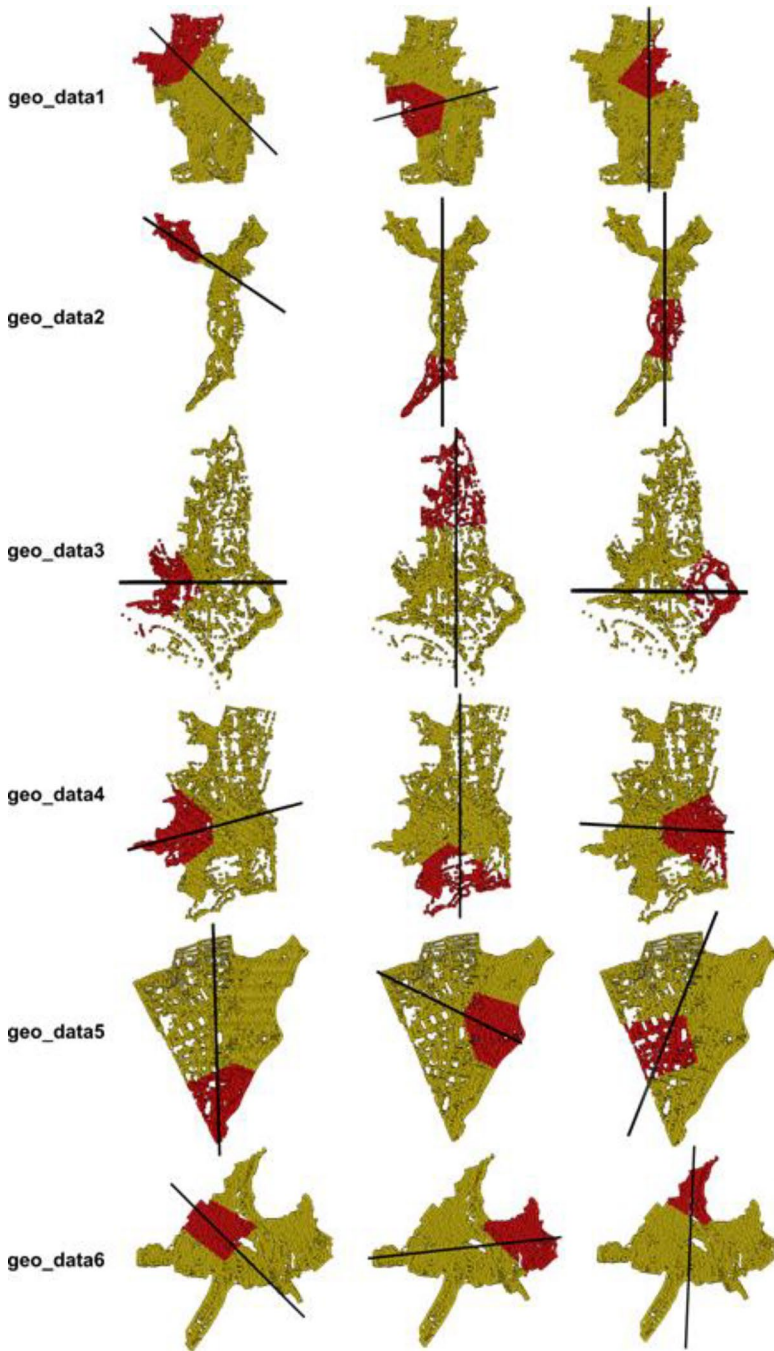
The results from the user tests are summarized in Table 1; Fig. 9 (a histogram). The collected user response shows how differently the particular users evaluate the approximate local symmetry. The average evaluation of pictures by the users is 2.3, which means that the users were only mildly satisfied with the suggested local symmetries. On the other hand, only a minority of results were evaluated as improper (evaluation 5). The tests did not show a dominance of either K-means or AABB methods but showed a slight dominance of the 1:0 setting of weights. It indicates that to get a human-convincing approximate local symmetry, it is better to concentrate the computation only on the local part of the shape and not consider the rest of the points, not even with a small weight. It is a rather unexpected result because we expected that the “global shape” of the data would influence the total impression.

Generally speaking, detecting symmetries by a human observer and computer are not in a fatal disagreement, but some differences exist. For example, when the computer program is asked to find an axis of symmetry, it finds some even in the data that human observer would not consider symmetrical. Naturally, a question arises whether there is some “range of symmetry” in which most people consider the data



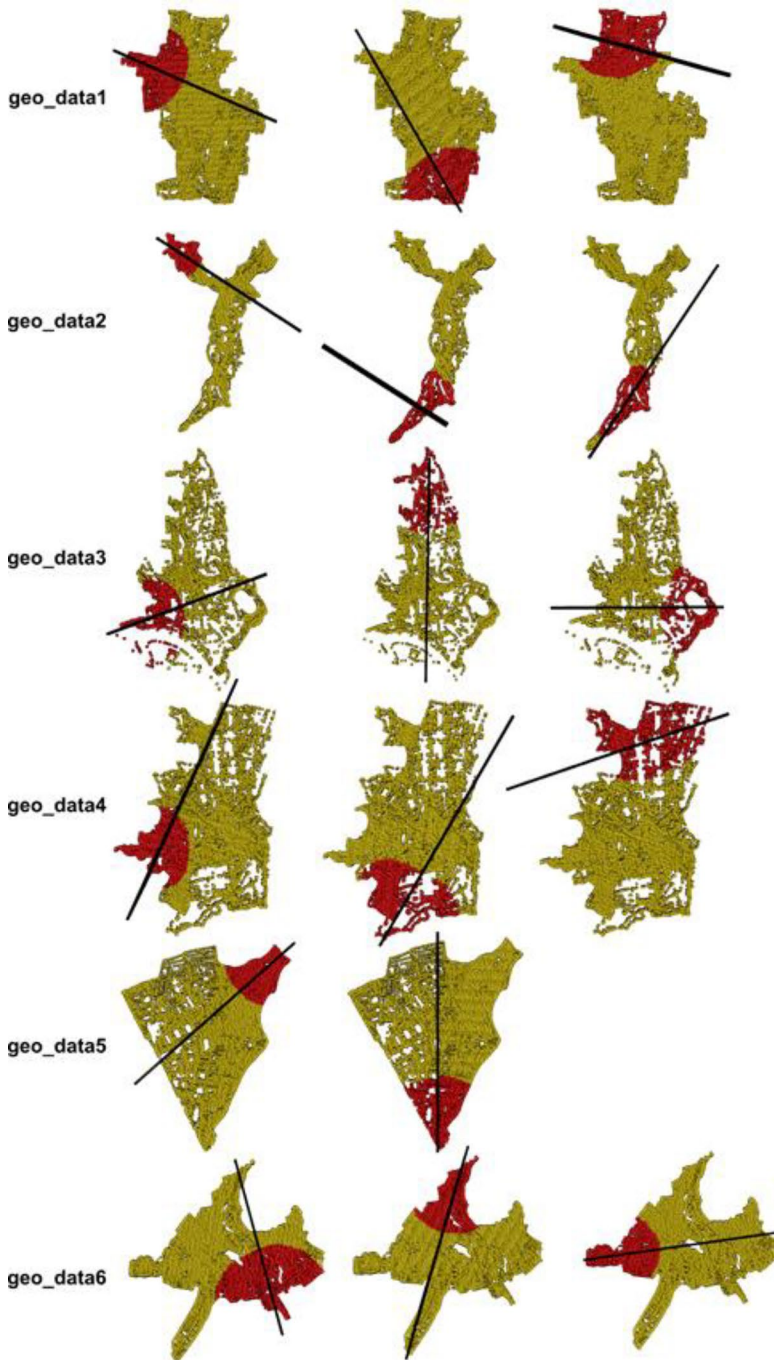
a)

Fig. 6 Resulting axes of local symmetry (black lines) for K-means method, **a** weights 0:1 (yellow points 0, red 1), **b** weights 1:5 (yellow points 1, red 5)



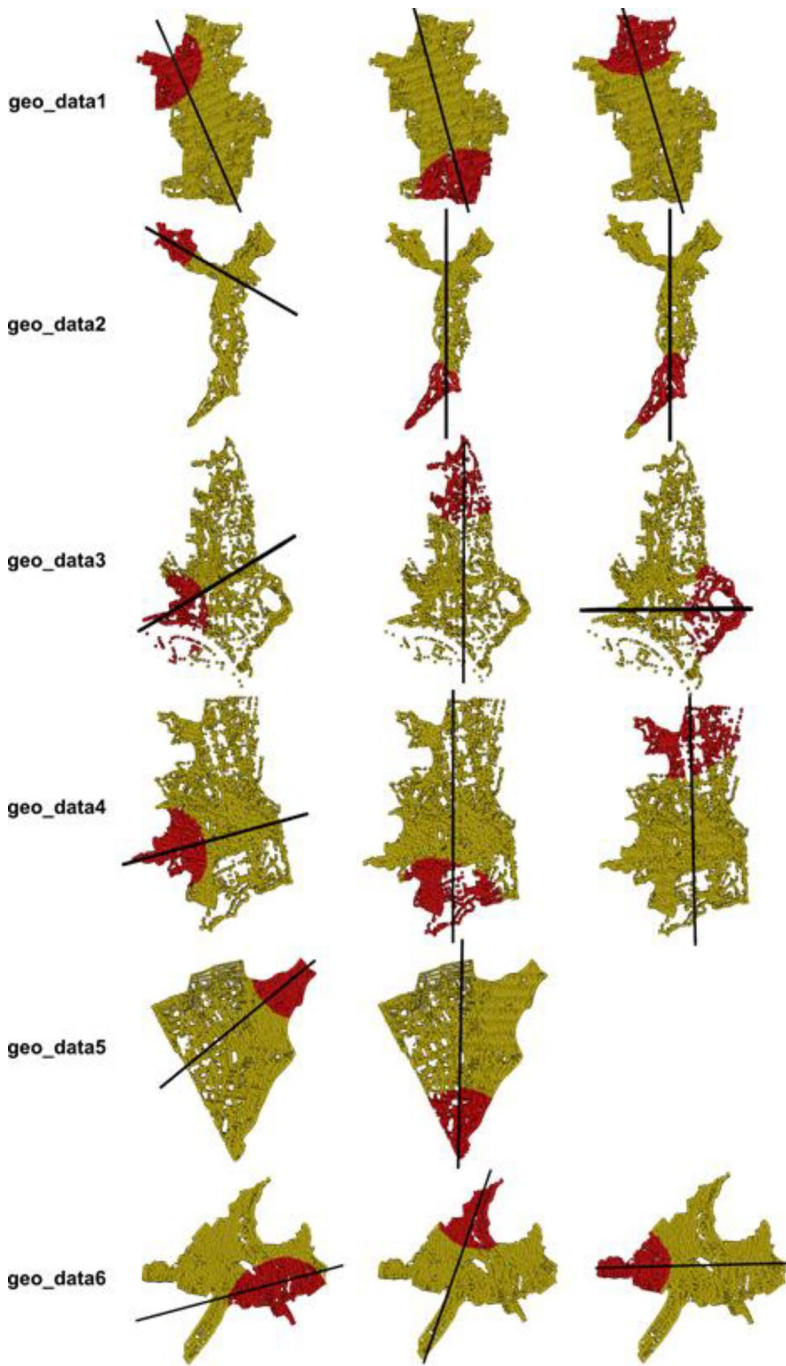
b)

Fig. 6 (continued)



a)

Fig. 7 Resulting axes of local symmetry (black lines) for AABB method, **a** weights 0:1 (yellow points 0, red 1), **b** weights 1:5 (yellow points 1, red 5)



b)

Fig. 7 (continued)

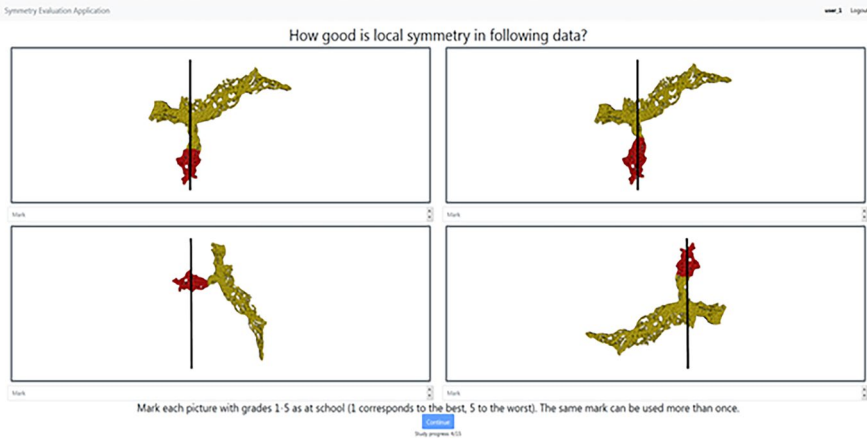


Fig. 8 A screenshot from the testing application

Table 1 User test results (number of users voting for evaluation 1–5 to the symmetry found by both algorithms, with two sets of weights for each algorithm)

| Mark/evaluation | K-means 1:0 | K-means 5:1 | Bounding Box 1:0 | Bounding Box 5:1 |
|-----------------|-------------|-------------|------------------|------------------|
| 1 | 85 | 72 | 100 | 84 |
| 2 | 90 | 94 | 80 | 74 |
| 3 | 65 | 64 | 48 | 69 |
| 4 | 29 | 28 | 35 | 30 |
| 5 | 21 | 32 | 27 | 33 |
| Average mark | 2.35 | 2.50 | 2.34 | 2.50 |

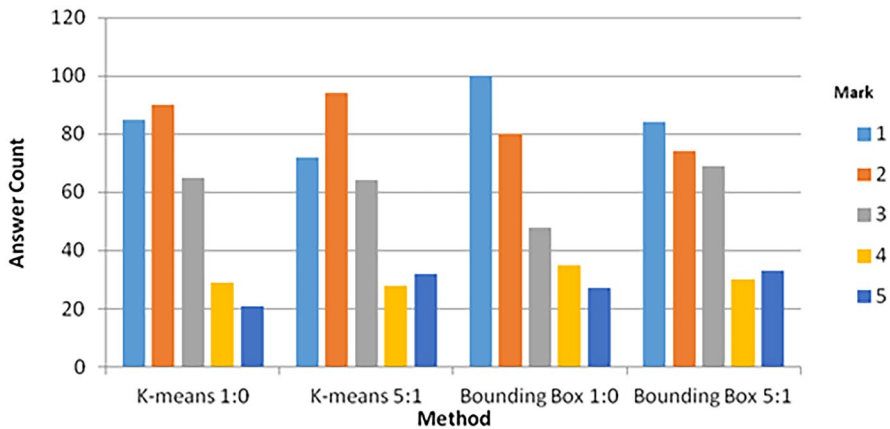


Fig. 9 Histogram of results obtained from the users (i.e., how many users voted for evaluation 1–5)

symmetrical. In our opinion, such research will have to be done first on artificially prepared testing data with various “degrees of asymmetry”; as much abstract as possible, with no resemblance to real life, with no mutual semantic relation, so that no kind of semantics (meaning) could influence the observer. Another interesting and valuable task for future work in human understanding of geodata symmetry would be to compare axes of symmetry drawn by humans with those generated automatically.

5 Conclusion

The question investigated in this paper is whether approximate local reflection symmetries in real data found by a computer are acceptable for human observers. In our case, we used geo data as the source of data is natural, and there is a wide variety in this type of data, which prevented us from using some data with very specific features (what may be, e.g., caused by properties of algorithms that generate the images). In such a way, we ensured that the results obtained were not influenced by some data specifics, which allowed us to conclude that the results obtained were not influenced by specific features in the images. To answer the research question, we proposed a new simple approximate local reflection symmetry detection and ran user tests.

The positive result of the research presented is that, in general, there is no dramatic difference between human and machine symmetry detection. This fact makes a reasonable assumption for computer-based support when humans detect approximate symmetry (to give hints to humans when they see no symmetry in the image).

Nevertheless, there were some cases where the differences were somewhat significant. The formal explanation for this phenomenon is relatively simple: symmetry in the image is detected using different principles by humans and computers. Detection of these reasons should be a goal of the follow-up research. It will be necessary to detect some specifics of human perception that cause differences in comparison with the image processing by a computer (why some differences emerge). When detecting these specifics - they will probably depend on the configuration of the image investigated, see, e.g., Gestalt laws – Wagemans et al. (2012) - it will be possible to adapt algorithms so that the computer symmetry detection will be closer to human symmetry detection.

Acknowledgements This research was supported by the Czech Science Foundation under research project 21–08009 K, the Czech Technical University in Prague, grant “RVO - Institutional support of research organization development”, No. RVO13000, and the Slovenian Research and Innovation Agency under research project N2-0181, and by Research Programme P2-0041.

Funding Open access publishing supported by the institutions participating in the CzechELib Transformative Agreement.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use

is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bairagi VK (2015) Symmetry-based biomedical image compression. *J Digit Imaging* 28(6):718–726. <https://doi.org/10.1007/s10278-015-9779-3>
- Bandyopadhyay S, Saha S (2013) Symmetry-based automatic clustering. Unsupervised classification. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-32451-2_7
- Bayer T, Kolingerová I, Potůčková M, Čábelka M, Štefanová (2023) An incremental facility location clustering with a new hybrid constrained pseudometric. *Pattern Recogn.* <https://doi.org/10.1016/j.patcog.2023.109520>
- Bigün J (1988) Pattern Recognition by Detection of Local Symmetries, Machine Intelligence and Pattern Recognition, North-Holland, Volume 7, pages 75–90
- Chalás I, Urbanová P, Kotulanová Z, Jandová M, Králík M, Kozlíková B, Sochor J (2014) Forensic 3D facial identification software (FIDENTIS). In: Proceedings of the 20th World Meeting of the International Association of Forensic Sciences
- Clode S, Rottensteiner, Kootsookos P, Zelniker E (2007) Detection and vectorization of roads from lidar data. *Photogram Eng Remote Sens* 73(5):517–535
- Ecins A, Fermuller C, Aloimonos Y (2017) Detecting reflectional symmetries in 3d data through symmetrical fitting. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 1779–1783
- Gao L, Zhang LX, Meng HY, Ren YH, Lai K, Kobbelt L (2021) PRS-Net: planar reflective symmetry detection net for 3D models. *IEEE Trans Vis Comput Graph* 27(6):3007–3018. <https://doi.org/10.1109/TVCG.2020.3003823>
- Gézero L, Antunes C (2019) Automated three-dimensional linear elements extraction from mobile lidar point clouds in railway environments. *Infrastructures* 4(3):46
- Girault JM (2024) Symmetry in signals: A new insight. *Entropy* 26(11):941. <https://doi.org/10.3390/e26110941>
- Hauagge DC, Snavel N (2012) Image matching using local symmetry features. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, pp. 206–213. <https://doi.org/10.1109/CVPR.2012.6247677>
- Hauert JH (2012) A symmetry detector for map generalization and urban-space analysis. *ISPRS J Photogrammetry Remote Sens* 74:66–77
- Hruda L (2021) Symmetry detection in geometric models, TR No. DCSE/TR-2021-01. <https://marowit.civ.zcu.cz/api/alfresco?id=5386efd3-025f-4083-90e5-1c9baf81dca1;1.0&download=0>
- Hruda L, Kolingerová I, Váša L (2022a) Robust, fast and flexible symmetry plane detection based on differentiable symmetry measure. *Visual Comput* 38:555–571. <https://doi.org/10.1007/s00371-020-02034-w>
- Hruda L, Kolingerová I, Váša L (2022b) Robust, fast and flexible symmetry plane detection based on differentiable symmetry measure, the supplementary material. <http://meshcompression.org/tvcj-2020>
- Hruda L, Kolingerová I, Podgorelec D (2023) Local reflectional symmetry detection in point clouds using a simple PCA-based shape descriptor. *VISIGRAPP (1: GRAPP) 2023*:52–63
- Ji P, Liu X (2019) A fast and efficient 3D reflection symmetry detector based on neural networks. *Multimed Tools Appl* 78:35471–35492. <https://doi.org/10.1007/s11042-019-08043-9>
- Jiang B, Liu X, Jia T (2011) Scaling of geographic space as a universal rule for map generalization. *Ann Assoc Am Geogr.* <https://doi.org/10.1080/00045608.2013.765773>
- Karakala R, Kaliamoorth P, Premachandran V (2019) Three-dimensional bilateral symmetry plane estimation in the phase domain. In: Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 249–256. <https://doi.org/10.1109/CVPR.2013.39>
- Kerber JM, Bokeloh M, Wand M, Seidel HP (2013) Scalable symmetry detection for urban scenes. *Comput Graphics Forum* 32(1):3–15

- Kootstra G, de Jong S, Schomaker LR (2009) Using local symmetry for landmark selection. In: Fritz M, Schiele B, Piater JH (eds) *Computer vision Systems. ICVS 2009. Lecture Notes in Computer Science*, vol 5815. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04667-4_10
- Korman S, Litman R, Avidan S, Bronstein A (2015) Probably approximately symmetric: fast rigid symmetry detection with global guarantees. *Comput Graphics Forum* 34:2–13. <https://doi.org/10.1111/cgf.12454>
- Krippendorf S, Syaeri M (2020) Detecting symmetries with neural networks. *Mach Learn Sci Technol* 2:015010
- Leng C, Zhang H, Li B, Cai G, Pei Z, He L (2019) Local feature descriptor for image matching: a survey. *IEEE Access* 7:6424–6434. <https://doi.org/10.1109/ACCESS.2018.2888856>
- Li BK, Johan H, Ye Y, Lu YY (2016) Efficient 3d reflection symmetry detection: a view-based approach. *Graph Models* 83:2–14
- Lightman A (2014) *The accidental universe: the world you thought you knew*. Knopf Doubleday Publishing Group
- Lipman Y, Chen X, Daubechies I, Funkhouser T (2010) Symmetry factored embedding and distance. In: *ACM SIGGRAPH 2010 papers*, pp. 1–12
- Martin GE (1996) *Transformation geometry: an introduction to symmetry*. Springer, pp. 28
- Martinet A, Soler C, Holzschuch N, Sillion FX (2006) Accurate detection of symmetries in 3D shapes. *ACM Transcations Graphics (TOG)* 25:439–464
- Mavridis P, Sipiran I, Andreadis A, Papaioannou G (2015) Object completion using k-sparse optimization. *Computer Graphics Forum* 34:13–21
- Mehafy MW (2020) The impacts of symmetry in architecture and urbanism: toward a new research agenda. *Buildings* 10(12):249. <https://doi.org/10.3390/buildings10120249>
- Mitra NJ, Guibas LJ, Pauly M (2006) Partial and approximate symmetry detection for 3d geometry. *ACM Trans Graphics (TOG)* 25(3):560–568
- Mitra NJ, Pauly M, Wand M, Ceylan D (2013) Symmetry in 3D geometry: extraction and applications. *Comput Graphics Forum* 32(6):1–23
- Murtagh F (2008) Symmetry in data mining and analysis: A unifying view based on hierarchy. *Proc Steklov Inst Math* 265. <https://doi.org/10.1134/S0081543809020175>
- Palmer SE (1985) The role of symmetry in shape perception. *Acta Psychol* 59(1):67–90
- Podgorelec D, Lukač L, Žalik B (2023) Reflection symmetry detection in Earth observation data. *Sensors* 23:7426
- Podolak J, Shilane P, Golovinsky A, Rusinkiewicz S, Funkhouser T (2006) A planar reflective symmetry transform for 3D shapes. *ACM Trans Graphics (TOG)* 25:549–559. <https://doi.org/10.1145/1141911.1141923>
- Qin T, Liu Z, Wang P, Li S, Guan X, Gao L (2020) Symmetry degree measurement and its applications to anomaly detection. *IEEE Trans Inf Forensics Secur* 15:1040–1055. <https://doi.org/10.1109/TIFS.2019.2933731>
- Schriebener D, Schmidt A, Vahrenkamp N, Asfour T (2016) Heuristic 3D object shape completion based on symmetry and scene context. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 74–81
- Shi Y, Huang J, Zhang H, Xu X, Rusinkiewicz S, Xu K (2020) SymmetryNet: learning to predict reflectional and rotational symmetries of 3D shapes from single-view. RGB-D images. *ArXiv abs/2008.00485*
- Simari P, Kalogerakis E, Singh K (2006) Folding meshes: hierarchical mesh segmentation based on planar symmetry. In: *Symposium on Geometry Processing*, vol. 256, pp. 111–119
- Sipiran I, Gregor R, Schreck T (2014) Approximate symmetry detection in partial 3D meshes. *Comput Graphics Forum* 33(7):131–140
- Sun C, Sherrah J (1997) 3D symmetry detection using the extended Gaussian image. *IEEE Trans Pattern Anal Mach Intell* 19:164–168. <https://doi.org/10.1109/34.574800>
- Sun QB, Huang WM, Wu JK (1998) Face detection based on color and local symmetry information. In: *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, pp. 130–135. <https://doi.org/10.1109/AFGR.1998.670937>
- Sundaram S, Sinha D, Groth M (2022) Recurrent connections facilitate symmetry perception in deep networks. *Sci Rep* 12:20931. <https://doi.org/10.1038/s41598-022-25219-w>
- Tu J, Sui H, Feng W, Sun K, Xu C, Han Q (2017) Detecting Building facade damage from oblique aerial images using local symmetry feature and the Gini index. *Remote Sens Lett* 8(7):676–685

- Uhrová K (2021) Zobecňená symetrie geometrických Dat (Generalized symmetry of geometric Data), diploma work, supervisor: Kolingerová I, Faculty of Applied Science. University of West Bohemia. (in Czech)
- Wagemans J, Feldman J, Gepshtein S, Kimchi R, Pomerantz JR, van der Helm PA, van Leeuwen C (2012) A century of Gestalt psychology in visual perception: II. Conceptual and theoretical foundations. *Psychol Bull.* 2012;138(6):1218–52. <https://doi.org/10.1037/a0029334>
- Wang H, Zhang W, Chen WY, Chen M, Yan K (2015) Semantic decomposition and reconstruction of compound buildings with symmetric roofs from lidar data and aerial imagery. *Remote Sens* 7(10):13945–13974
- Zabrodsky H (1990) Symmetry - a review. Technical report, Department of Computer Science, The Hebrew University of Jerusalem, 1990
- Zabrodsky H, Algom D (1994) Continuous symmetry: a model for human figural perception. *Spat Vis* 8(4):455–467. <https://doi.org/10.1163/156856894x00116>
- Žalik KR, Žalik B (2009) A sweep-line algorithm for Spatial clustering. *Adv Eng Softw* 40(6):445–451. <https://doi.org/10.1016/j.advengsoft.2008.06.003>
- Žalik B, Strnad D, Podgorelec D, Kolingerová I, Nerat A, Lukač N, Kohek Š, Lukač L (2023) Geometric shape characterisation based on a multi-sweeping paradigm. *Symmetry* 15:1212. <https://doi.org/10.3390/sym15061212>
- Zhou Q, Jacobson A (2016) Thingi10k: A dataset of 10,000 3d-printing models. *ArXiv Preprint arXiv:1605.04797*

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.