

Finding the Shear Reflection Symmetry Plane in a 3D Point Cloud

Vítek Poór¹^a, Ivana Kolingerová¹^b and Damjan Strnad²^c

¹Department of Computer Science and Engineering, University of West Bohemia,
Technická 8, 306 14 Plzeň, Czech Republic

²Department of Computer Science, Faculty of Electrical Engineering and Computer Science, University of Maribor,
Koroška cesta 46, 2000 Maribor, Slovenia

Keywords: Symmetry, Reflection, Shearing, Computer Graphics.

Abstract: Many objects, namely man-made ones, show signs of various types of symmetry. The most common type perceived by humans is reflection symmetry to some plane. When detecting the symmetry for geometric models, the existing algorithms look for orthogonal reflection symmetry. However, the models can be sheared, therefore, algorithms detecting shear reflection symmetry would be useful. In this paper, we propose an algorithm for detecting the plane of shear reflection symmetry in a 3D point cloud on condition that the shear was done in one of the coordinate axes.

1 INTRODUCTION

Symmetry is a distinctive visual feature of geometric objects, enabling their faster human understanding. Mathematically, it is typically defined as a geometric transformation different from the identity, mapping the object into itself. The most often used transformation is planar reflection, mirroring the object by a plane of symmetry, and many methods for its computer detection have been developed. However, the used reflection is orthogonal, although managing more general directions of reflection could be useful to handle objects that are symmetric in a more general sense, e.g., an object and its skewed copy.

When detecting the plane of symmetry for computer geometric models, we work with approximate symmetry, since we cannot rely on perfect symmetry in real 3D data. Therefore, it is reasonable to propose symmetry detection algorithms that work with noisy or incomplete data.

This paper makes the first step to handling one variant of non-orthogonal reflection symmetry, a shear reflection symmetry. The proposed algorithm handles 3D point clouds, possibly noisy or incomplete, sampled from the surface of a geometric object. The input data are supposed to have been sheared in the x -direction, by an unknown angle. The output is

the best fitting plane of shear reflection symmetry and the shear angle to the x -axis.

1.1 Background

The perfect symmetry of the given 3D points of an object X can be understood as a geometric transformation T such that $T(X) = X$, i.e., the object X is invariant to the transformation T . The approximate symmetry can then be understood as a geometric transformation T such that $T(X)$ approximately matches X .


A common type of symmetry is reflection symmetry, where the transformation T is a reflection over a given plane. Let the plane be described by the signed distance from the origin d and the normal vector \mathbf{n} . Then the orthogonal reflection of the point \mathbf{p} onto its image \mathbf{r} through the plane (\mathbf{n}, d) is given by


$$\mathbf{r} = \mathbf{p} - 2 * (\text{dot}(\mathbf{n}, \mathbf{p}) - d) * \mathbf{n}, \quad (1)$$


where dot is the scalar product of two vectors.

Naturally or artificially skewed data can be described by an affine transformation that shifts each point in a certain direction by the value of its distance from a given line parallel to that direction. This transformation is often called shear mapping, shear transformation or just shearing. The equation of the general affine transformation, mapping the source vector $\mathbf{x} = [x, y, z]^T$ by the transformation matrix \mathbf{A} into the destination vector $\mathbf{x}' = [x', y', z']^T$, is as follows:

$$\mathbf{x}' = \mathbf{A}\mathbf{x}. \quad (2)$$

^a <https://orcid.org/0009-0004-4489-2837>

^b <https://orcid.org/0000-0003-4556-2771>

^c <https://orcid.org/0000-0003-4468-0290>

In 3D space, shearing can be done in three directions. The corresponding transformation matrices have the following forms:

$$\text{shearing in } x\text{-direction: } A = \begin{bmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ b & 0 & 1 \end{bmatrix}, \quad (3)$$

$$\text{shearing in } y\text{-direction: } A = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & b & 1 \end{bmatrix}, \quad (4)$$

$$\text{shearing in } z\text{-direction: } A = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}, \quad (5)$$

where a and b are coefficients of the shear; shearing in the x -direction affects all axes except the x -axis, etc. It is possible to assemble shearing in more axes, however, in this work, only the shear matrix (4) is used and, what is more, $b = 0$.

The coefficients are derived from the angle at which the object is sheared. We call this angle the shear angle and the coefficients are its tangent. For example, if the object is to be sheared in the y -direction at an angle of 45° only due to the influence of the x -axis, then the matrix (4) with the coefficients $a = \tan(45)$ and $b = 0$.

1.2 Related Work

Symmetry detection is widely studied in various fields including compression (Simari et al., 2006), object reconstruction (Sipiran, 2017) and alignment (Chaouch and Verroust-Blondet, 2009), re-meshing (Podolak et al., 2007), symmetrical editing (Martinet et al., 2006), facial image analysis (Mitra and Liu, 2004). Due to the importance of this topic, various types of symmetry were addressed, such as perfect, approximate, global, local, extrinsic, intrinsic, rotational, and reflection; the detection methods operate namely on images and 3D points. An overview is provided in (Mitra et al., 2013).

As reflection symmetry is the most important, there are many methods for its detection. Most of them have some limitations; e.g., they detect only planes passing through a reference point (for example, the object centroid) (Sun and Sherrah, 1997; Martinet et al., 2006; Li et al., 2016). These methods are suitable for determining the global symmetry of the entire object but not its local symmetry where different reference points are needed. The method in (Schiebener et al., 2016) does not have this limitation and is thus suitable also for local symmetry but additional knowledge is needed, such as the position

from which the object is scanned. A similar method is (Ji and Liu, 2019), which needs an additional neural network training dataset. There are a few methods that impose no limitations on the input data, such as in (Hruda et al., 2022).

Shear is indirectly used in the search for symmetry in images, as the projective transformation distorts the objects in the scene and they appear skewed. Several algorithms have been presented to solve this problem, e.g., (Gross and Boulton, 1994; Bruckstein and Shaked, 1998; Friedberg, 1986; Cham and Cipolla, 1995). These algorithms are based on the extraction of features from the shape and either take into account the entire contour and are insensitive to noise, or consider locally defined characteristic features; it brings instability and sensitivity to noise.

As far as we know, there is no direct method for shear symmetry in a 3D point cloud. However, methods for object registration can be applied - we can see symmetry detection as a special case of registration where both input objects are the same. Among the registration algorithms, there are also specific algorithms for affine transformations (Ji et al., 2017; Shu et al., 2021). However, these algorithms need a proper adjustment so that their output is a detected plane of symmetry and they are unnecessarily complex for solving the symmetry detection problem.

Our work focuses on global, approximate, shear reflection symmetry in a 3D point cloud, sampled from the surface of the object. For easier understanding, the algorithm is first described in a simplified version, for a fixed shear angle, in Section 2. An extension of the algorithm for an arbitrary shear angle is presented in Section 3. The limitation of the algorithm is that only shear along the x -axis is taken into account. The case of an unknown shear direction could be solved by triple repetition of the algorithm. However, a general shear in a more complicated direction, combined from transformations in more axes, is out of scope of the algorithm. Such shears are complicated even to be perceived correctly and they are ambiguous in the sense that the results could have been achieved by more than one series of transformations.

2 THE SIMPLIFIED ALGORITHM FOR A FIXED SHEAR ANGLE

The input of the algorithm is a set of 3D points from the surface of an object, the output is the equation of the plane of shear symmetry together with the shear angle. A plane (\mathbf{n}, d) is described by its normal \mathbf{n}

and its distance d from the origin. The shear angle tells us the angle of transformation of the shear object in a certain direction (compared to the untransformed object). The version of the algorithm presented in this section works with a fixed direction and angle (i.e., the fixed shear angle is the same as the shear angle at the output). The shear reflection symmetry plane detection algorithm is composed of three main steps:

1. Generating orthogonal and shear planes,
2. Evaluation of all planes,
3. Selection of the best plane.

A detailed description of the algorithm steps is beyond the scope of this text.

2.1 Generating Planes

Planes are generated directly from the input point cloud. All pairs of points are processed, and each pair generates two planes. Let the points p and q be the currently processed pair of points. The first plane reflects orthogonally the point p to the point q (see Equation 1). We need this plane for cases where the detected reflection symmetry plane is not sheared. The second plane, called the shear plane, reflects at a shear angle. In addition, the input of the algorithm contains the shear vector *shear*, the first two components of which indicate the shear direction and the third component the shear angle.

Shear reflection through the original plane is achieved by constructing an auxiliary plane whose normal vector ns is in the direction of the input vector *shear* and its distance from the origin is determined by the distance of the intersection x of the original plane with the line constructed by the input point p and the ns direction (see Fig. 1). The term *original plane* refers to the plane currently being processed by the algorithm.

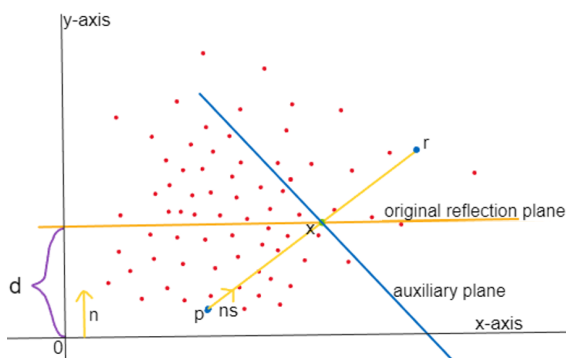


Figure 1: An example of constructing an auxiliary plane to achieve shear reflection.

2.2 Evaluation of all Planes

The next step of the algorithm is the evaluation of the generated planes. For each plane, the evaluation is based on the distances of the mirrored points from the original points. First, a plane is constructed from the auxiliary vector, and then a measure is evaluated for it. This evaluation process is performed for all points of the input set over all generated planes. The overall evaluation of one plane is given by the sum of partial evaluations.

Finally, the partial evaluation of the given plane for one given point from the input set of points is calculated. Through the shear plane constructed as described in the previous text, the point p is reflected to the point r orthogonally via Equation 1. From the thus reflected point r , the distances of all points of the input set are calculated. The partial evaluation of the input plane for the input point p is calculated by the function minimum which returns the minimum value from the distances of all points to point p .

2.3 Selection of the Best Plane

The selection of the plane with the best evaluation is done by the ascending rating of the resulting evaluations associated with the planes. The plane with the lowest rating is taken as the best. In other words, the mirror of points obtained by shear reflection through the given plane has the smallest distance of points from the original points.

However, the algorithm can return more planes since it has the entire list of evaluated planes at its disposal. More planes than one may be desirable for objects where the plane of symmetry is not clearly visible to humans. Such a selection is then made by the user, who must take into account the value of the evaluation.

2.4 Optimization

The proposed algorithm has been so far explained as brute force. To achieve a reasonable runtime, various optimization techniques can and should be incorporated. One of the techniques with the biggest impact is input data reduction. A 3D spatial grid with a fixed cell size is built over the input data points. All points in individual cells are then averaged. The points created in this way are then used in the next steps of the algorithm. The number of points is directly proportional to the number of grid cells. The input points are normalized to a cube of unit edge and shifted so that the centroid of the object is at the origin of the world coordinate system.

The biggest bottleneck of the algorithm is the evaluation of the planes. Optimization is done both to reduce the set of planes to be evaluated and for the evaluation itself. One can imagine that the generated planes in the first step are many, especially for large data sets. As part of this step, the conditions for adding a new plane to an existing data structure are established. The distance of the plane from the origin and the deviation of the direction of the normal from the fixed directions are verified. Assuming that the resulting plane passes through the origin (or its small neighborhood), we keep only planes with the distance from the origin smaller than the chosen threshold e_d . Similarly, we can also limit the normal direction of the plane, with a known shear it is likely that the resulting plane will be somewhere within the angle e_a from the direction of shear or the orientation of world coordinates.

Another significant optimization is performed in the evaluation of planes. The experiments show that a lot of time is devoted to the evaluation of planes that end up with a worse measure than the best planes. Since the resulting measure of one plane is given as the sum of partial measures of all points, it is possible to terminate the evaluation process of the current plane on the base of additional knowledge, such as the best evaluation received so far.

3 THE FULL ALGORITHM FOR AN ARBITRARY SHEAR ANGLE

The algorithm described so far worked with additional knowledge of the shear angle (obtained at the input of the algorithm). This section describes a generalization of the algorithm by removing the limitation of a previously known shear angle. Thus, the required input of the algorithm is only a set of 3D points. The current limitation of the algorithm to the x -axis remains.

3.1 The Proposed Extension

The problem of finding an unknown shear angle in one axis corresponds to the problem of finding an extreme in the graph of plane evaluations as a function of shear angle, see an example in Figure 2. For time reasons, we want to avoid the computation of this graph and still find the extreme. In order to avoid computation in the entire interval of angles, an initial estimate of the shear angle is needed.

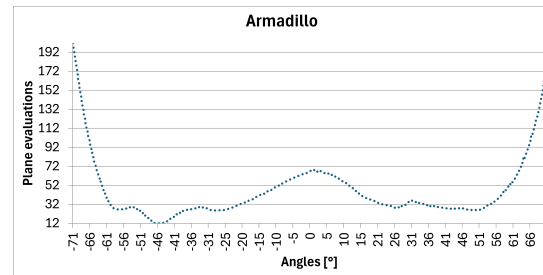


Figure 2: Planes evaluation for various shear angles over the Armadillo object. Microsoft Excel was used as the graph generator.

3.2 Shear Angle Estimation

One can imagine the estimate of the shear angle as the angle between the y -axis and a straight line dividing the object vertically in two when projected onto the xy -plane. The straightforward approach is the use of a pre-created 3D grid. In particular, it concerns the base and ceiling cells. From these two sets of cells, two points are determined as the centroids of the points contained in these sets. These two points construct a straight line that passes through the object. When projecting such a straight line into 2D (xy -plane in our case), the angle between the straight line and the y -axis indicates the skew of the grid. This angle is also the desired estimate. A schematic is shown in Figure 3, with an example on a real data set shown in Figure 4. The estimate is then used to construct an initial interval for finding the resulting shear angle.

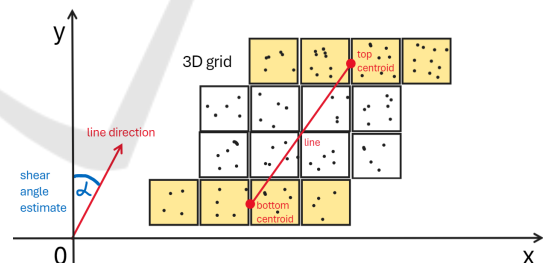


Figure 3: Sketch of the shear angle estimation construction process based on the bottom and top points of the 3D grid cells. For simplicity, projection to the xy -plane is used.

3.3 Finding Shear Angle in Interval

With a known estimate of the shear angle, the search of the optimum can be limited to only a small sub-interval of angles. First, a total of five values are determined, from which the search sub-interval is then constructed. The five values are a set formed around the estimate. For the estimate α_e and for the neighborhood coefficient b , the set is composed as

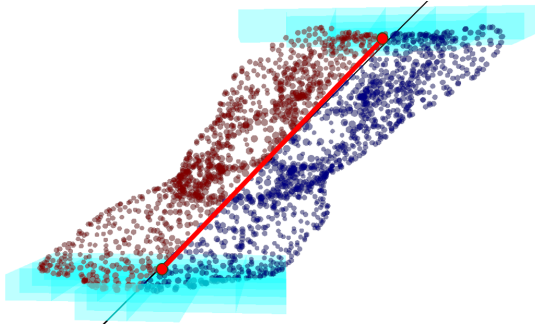


Figure 4: Shear angle estimate on the Lion data with the angle estimate (red line) and the plane found by the algorithm (black line). For simplicity, only the bottom and top cells of the 3D grid are shown, which are used to calculate the centroids and construct the line.

$$\{-(2*b) + \alpha_e, -b + \alpha_e, \alpha_e, \alpha_e + b, \alpha_e + (b*2)\}. \quad (6)$$

Subsequently, a measure is calculated for these five values using our detection algorithm. A search sub-interval is constructed from the two values with the best measure. This process is repeated for $-\alpha_e$. At the end, a total of two search sub-intervals are available, one for a positive estimate and one for a negative estimate. A binary search is used over these two intervals. The two best obtained values are compared and the best value is declared as the resulting shear angle.

4 EXPERIMENTS AND RESULTS

First, in Section 4.1, the basic experiments and results for the simplified proposed method with a fixed shear angle are presented. These experiments focus on the correctness of the resulting plane and the calculation time. Further in Section 4.2 there are experiments and results for the extended method of finding the shear angle. Here, the main subject is comparing the results of the extended method against the best possible ones obtained from previous experiments.

4.1 Experiments and Results for Simplified Method

Figure 5 shows objects with their sheared symmetry planes detected by the proposed method. Objects are displayed so that the detected plane (indicated by a line) is perpendicular to the plane of projection. They were taken from various datasets (Fang et al., 2008; Levoy et al., 2005; Shilane et al., 2004). The results except for Bunny look correct in visual inspection. More detailed analysis is beyond the scope of this text.

Since we do not know of another method for shear symmetry detection, we cannot perform a comparison and instead compare the best detected shear symmetry plane with the best detected reflection symmetry plane that is obtained for un-sheared data. We mainly experimented with the plane evaluation measurement across different angles and the algorithm run time.

As a reference method for the detection of orthogonal symmetry, we have chosen (Hruda et al., 2020). This method does not require additional information for input data. As an experiment, we took the output of the reference method for an object that was not sheared. Then we sheared the result plane according to the best angle obtained by our algorithm. We obtained the best shear angle by re-running our algorithm for angles ranging from -90 to 90 degrees with the step equal to 0.1 and selecting the best result. The comparison is based on the angle difference between our detected plane and the plane additionally sheared according to our obtained shear angle. Table 1 shows the results for all objects. The biggest difference can be seen with the Bunny object. This is mainly due to the selection of the best angle.

If we detect a shear symmetry plane in the data, where we know the shear angle, then we can assume that the detected plane will be sheared at an angle opposite to the angle of the sheared data. We performed an experiment that detects a shear plane with different fixed shear angles for one specific object, Armadillo (see Figure 5 (b)), sheared in the x -axis at an angle of 45 degrees. Figure 2 shows the results of individual measurement values for different angles (a smaller value is better). One can see that the best evaluation is around an angle of -45 degrees, which was the assumption of the experiment.

Computation time for the tested objects can be seen in Table 2. Different sizes of simplification depending on the total number of points are determined by the chosen simplification algorithm and the shape of the object.

4.2 Experiments and Results for Proposed Method

Table 3 compares the best obtained shear angle (as in Table 1) with the shear angle found using the extended algorithm. Data is generated for every single instance of the Armadillo object (see Figure 5 (b)) that has been sheared over the interval $(-60, 60)$. Only a few representative samples were selected.

Table 4 summarizes the error rate statistics of the proposed algorithm extension compared to the best found values. This experiment extends the previous Table 3 by all previously used objects.

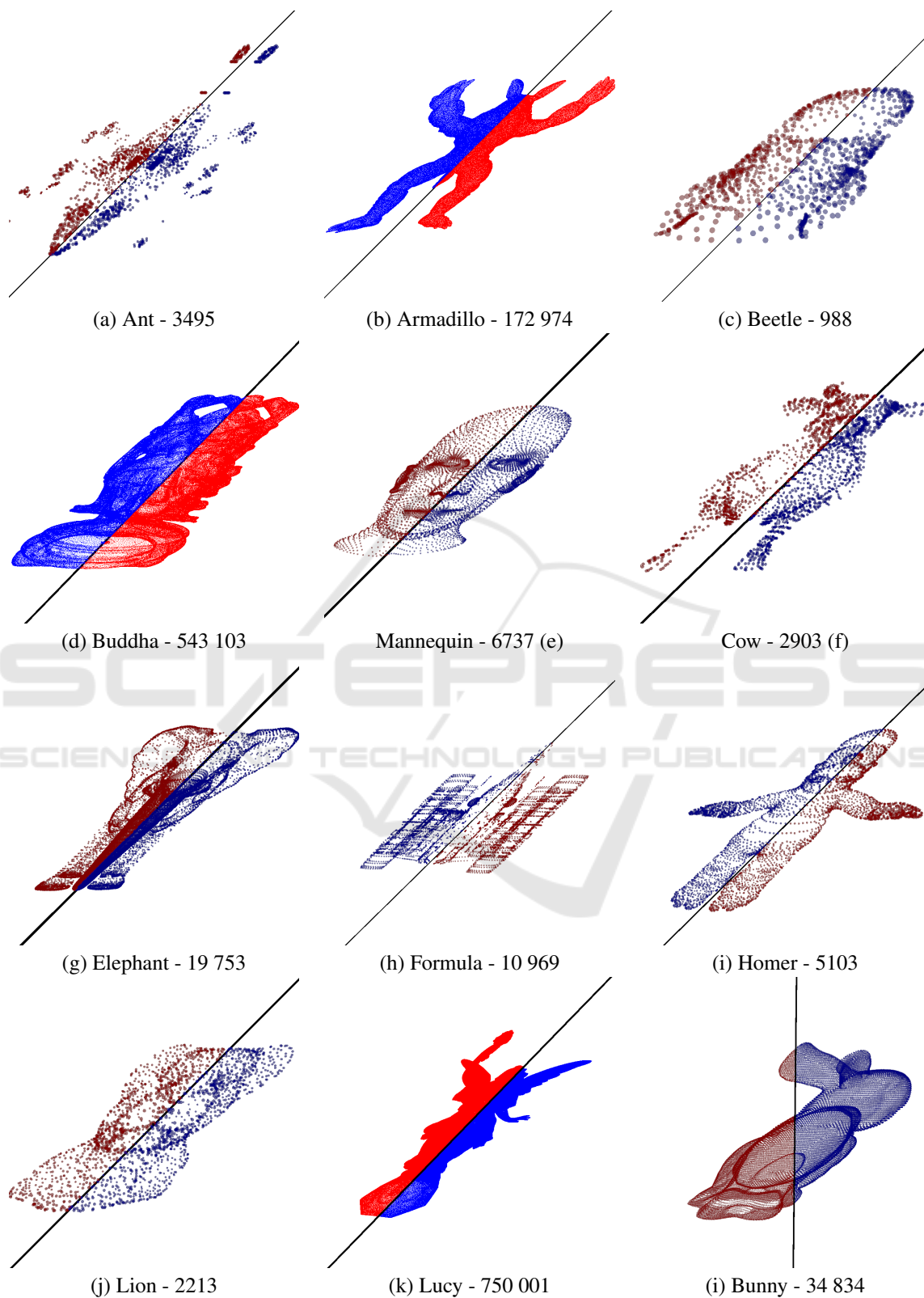


Figure 5: Several objects with their symmetry planes detected using the proposed method. The number indicates the number of 3D points in the point cloud.

Table 1: Comparison of angle differences between the plane normals of the reference method and the proposed method. The reference plane is sheared according to the best angle we obtained. Then both shear planes are compared. The distance from the origin d is not shown as the compared planes pass through the centroid of the object.

Object	Reference plane	Shear Angle [°]	Reference Shear Plane	Resulting Plane	Angle Diff. [°]
Ant	(1.00, 0.00, 0.01)	44.9	(0.71, -0.70, 0.00)	(-0.71, 0.71, -0.01)	0.60
Armadillo	(1.00, 0.00, 0.01)	45.5	(0.70, -0.71, 0.00)	(0.70, -0.71, 0.00)	0.21
Beetle	(1.00, 0.00, 0.01)	45.1	(0.71, -0.70, 0.00)	(0.71, -0.71, 0.00)	0.19
Buddha	(-1.00, 0.00, 0.00)	44.5	(-0.71, 0.70, 0.00)	(0.71, -0.70, 0.00)	0.18
Bunny	(-0.99, 0.01, 0.17)	29.6	(-0.84, 0.49, 0.23)	(1.00, -0.01, -0.01)	32.26
Cow	(-1.00, 0.01, 0.01)	45	(-0.70, 0.71, 0.01)	(-0.70, 0.72, 0.01)	0.31
Elephant	(1.00, 0.00, -0.01)	45.1	(0.71, -0.71, 0.00)	(-0.71, 0.71, 0.01)	0.58
Formula	(1.00, -0.01, 0.00)	44.4	(0.71, -0.70, 0.00)	(0.71, -0.71, 0.00)	0.17
Homer	(-1.00, -0.01, -0.02)	45	(-0.71, 0.70, -0.02)	(0.71, -0.70, 0.00)	1.32
Lion	(1.00, -0.01, 0.00)	44.9	(0.70, -0.71, 0.00)	(-0.71, 0.71, -0.01)	0.63
Lucy	(-1.00, -0.04, 0.03)	45.5	(-0.73, 0.69, 0.04)	(-0.72, 0.69, 0.01)	1.91
Mannequin	(-1.00, 0.00, 0.00)	44.9	(-0.71, 0.71, 0.00)	(-0.71, 0.71, 0.00)	0.28

Table 2: Computation time for one fixed shear angle over all objects. Times are measured for simplified objects.

Object	Points	Points (simpl.)	Time [ms]
Ant	3495	162	508
Armadillo	172 974	281	1376
Beetle	988	283	1569
Buddha	543 103	361	2661
Bunny	34 834	342	2253
Cow	2903	253	1217
Elephant	19 753	272	1260
Formula	10 969	372	2981
Homer	5103	335	2389
Lion	2213	305	1797
Lucy	750 001	247	1301
Mannequin	6737	315	2012

Table 3: Comparison between the best obtained shear angle and the found shear angle for the Armadillo object. The individual data angles represent the partial shear instances of the Armadillo object.

Data Angle	Shear Angle [°]	Found Shear Angle [°]	Angle Diff. [°]
-59	58.0	58.5	00.5
-50	49.0	46.9	02.1
-40	40.0	38.6	01.4
-30	29.0	29.4	00.4
-20	19.0	19.3	00.3
-10	09.0	09.3	00.3
0	00.0	-00.4	00.4
10	-11.0	-11.2	00.2
20	-21.0	-21.0	00.0
30	-31.0	-30.5	00.5
40	-41.0	-40.4	00.6
50	-51.0	-51.1	00.1
59	-60.0	-59.6	00.4

5 CONCLUSION AND FUTURE WORK

We proposed a new method for the detection of shear reflection symmetry planes. We verified the method on a set of objects. Currently, the method works with shearing in one direction of the x -axis. Another limitation is the chosen level of simplification of the input point cloud, which has a direct impact on the result and calculation. Future work will focus on removing the limitation to include an adaptive choice of point cloud simplification level such that the smallest possible data still gives correct results. At the same time, another interesting challenge for future work is the limitation of the fixed shearing axis direction.

ACKNOWLEDGEMENTS

This research was supported by the Czech Science Foundation under research project 21-08009K, the Slovene Research and Innovation Agency under research project N2-0181, Research Programme P2-0041; V. Poór was also supported by the Ministry of Education, Youth and Sports under the Students Research project SGS-2022-015.

Table 4: Summary statistic error rate over all instances of the used models over the interval (-60, 60). The acquisition error rate is based on the experiment in Table 3, i.e. the difference between the best obtained angle and the found angle over the shear data interval.

Object	Min. Diff. [°]	Max. Diff. [°]	Avg. Diff. [°]	Med. Diff. [°]
Ant	0.00	1.00	0.25	0.00
Armadillo	0.00	3.40	0.54	0.40
Beetle	0.00	1.40	0.45	0.30
Buddha	0.00	6.10	0.99	0.50
Bunny	0.00	4.60	0.79	0.50
Cow	0.00	3.50	0.43	0.40
Elephant	0.00	1.00	0.25	0.10
Formula	0.00	2.40	0.53	0.40
Homer	0.00	2.00	0.32	0.20
Lion	0.00	1.80	0.50	0.40
Lucy	0.00	3.50	0.60	0.50
Mannequin	0.00	1.50	0.40	0.40

REFERENCES

- Bruckstein, A. M. and Shaked, D. (1998). Skew-symmetry detection via invariant signatures. *Pattern Recognition*, 31(2):181–192.
- Cham, T.-J. and Cipolla, R. (1995). Symmetry detection through local skewed symmetries. *Image and Vision Computing*, 13(5):439–450.
- Chaouch, M. and Verroust-Blondet, A. (2009). Alignment of 3d models. *Graphical Models*, 71(2):63–76.
- Fang, R., Godil, A., Li, X., and Wagan, A. (2008). A new shape benchmark for 3d object retrieval. In *International Symposium on Visual Computing*, pages 381–392. Springer.
- Friedberg, S. A. (1986). Finding axes of skewed symmetry. *Comput. Vision Graphics Image Process.*, 34:138–155.
- Gross, A. D. and Boulton, T. E. (1994). Analyzing skewed symmetries. *Int. J. Comput. Vision*, 13:91–111.
- Hruda, L., Kolingerová, I., and Váša, L. (2020). Robust, fast and flexible symmetry plane detection based on differentiable symmetry measure: Supplementary material. Available at <http://meshcompression.org/tvcj-2020>.
- Hruda, L., Kolingerová, I., and Váša, L. (2022). Robust, fast and flexible symmetry plane detection based on differentiable symmetry measure. *Visual Comput.*, 38:555–571.
- Ji, P. and Liu, X. (2019). A fast and efficient 3d reflection symmetry detector based on neural networks. *Multimedia Tools and Applications*, 78(24):35471–35492.
- Ji, S., Ren, Y., Ji, Z., Liu, X., and Hong, X. (2017). An improved method for registration of point cloud. *Optik*, 140:451–458.
- Levoy, M., Gerth, J., Curless, B., and Pull, K. (2005). The stanford 3d scanning repository. Available at <http://www.graphics.stanford.edu/data/3Dscanrep/>.
- Li, B., Johan, H., Ye, Y., and Lu, Y. (2016). Efficient 3d reflection symmetry detection: A view-based approach. *Graphical Models*, 83:2–14.
- Martinet, A., Soler, C., Holzschuch, N., and Sillion, F. X. (2006). Accurate detection of symmetries in 3d shapes. *ACM Transactions on Graphics (TOG)*, 25(2):439–464.
- Mitra, N. J., Pauly, M., Wand, M., and Ceylan, D. (2013). Symmetry in 3d geometry: Extraction and applications. *Computer Graphics Forum*, 32(6):1–23.
- Mitra, S. and Liu, Y. (2004). Local facial asymmetry for expression classification. In *CVPR*.
- Podolak, J., Golovinskiy, A., and Rusinkiewicz, S. (2007). Symmetry-enhanced remeshing of surfaces. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing (SGP '07)*, pages 235–242. Eurographics Association.
- Schiebener, D., Schmidt, A., Vahrenkamp, N., and Asfour, T. (2016). Heuristic 3d object shape completion based on symmetry and scene context. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 74–81. IEEE.
- Shilane, P., Min, P., Kazhdan, M., and Funkhouser, T. (2004). The princeton shape benchmark. In *Shape Modeling Applications, 2004. Proceedings*, pages 167–178. IEEE.
- Shu, Q., He, X., Wang, C., Yang, Y., and Cui, Z. (2021). Fast point cloud registration in multidirectional affine transformation. *Optik*, 229:165884.
- Simari, P., Kalogerakis, E., and Singh, K. (2006). Folding meshes: Hierarchical mesh segmentation based on planar symmetry. In *Symposium on Geometry Processing*, volume 256, pages 111–119.
- Sipiran, I. (2017). Analysis of partial axial symmetry on 3d surfaces and its application in the restoration of cultural heritage objects. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2925–2933.
- Sun, C. and Sherrah, J. (1997). 3d symmetry detection using the extended gaussian image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):164–168.