DOI: 10.6977/IJoSI.202209_7(3).0001
S. Kodama / Int. J. Systematic Innovation, 7(3), 1-15 (2022)

Learning and recognizing three-dimensional shapes by a neural network using solid angles

Satoshi Kodama[1*]
[1]Department of Information Technology, Faculty of Engineering,
International Professional University of Technology in Tokyo, Japan
[*] Corresponding author E-mail: kodama.satoshi@iput.ac.jp

(Received 10 April 2020; Final version received 22 February 2021; Accepted 23 August 2022)

Abstract

Three-dimensional (3D) shapes differ from two-dimensional (2D) shapes in terms of the amount of data that cat be acquired for each shape. In addition, the information that can be obtained from a 3D shape varies greatly depending on the viewing angle and posture, and there is currently no universal countermeasure for this problem. Therefore, it is difficult to acquire the level of features necessary for machine learning. To learn and recognize 3D shapes, learning approaches using images from various angles, techniques using normal vectors, and approaches based on the acquisition of the overall structure via voxelization have been studied thus far. However, these methods are not always effective because they complicate the preprocessing of data required for learning. In this paper, the author proposes a method using solid angles as a new quantitative feature for learning and recognition. The solid angle is a 3D angle corresponding to the plane angle of a 2D shape; when a point is fixed, a constant value can be obtained regardless of the posture of the object. In addition, although the calculations required to obtain this value are intensive and time consuming, they can be performed in a relatively simple manner. In this study, primitive shapes are learned and recognized using solid angles as a quantitative feature. As a result, the author demonstrates that after learning using a neural network, this method can appropriately recognize a given shape.

Keywords: Neural networks, Shape recognition, Shape registration, Solid angle.

1. Introduction

Unlike two-dimensional (2D) shapes, three-dimensional (3D) shapes inevitably have a substantial amount of characteristic data. Moreover, because one must perform appropriate processes for each encounter depending on the rotation of the object and the viewing angle, complicated processing is required, unlike that required for a 2D shape. Therefore, machine learning tends to be difficult for 3D shapes, in constant to 2D shapes (AI-SCHOLAR, 2018; Cohen et al., 2018; Fang et al., 2015; Mescheder et al., 2018).

In general, methods for learning and recognizing 3D shapes include learning approaches using images from various angles, techniques using normal vectors, and approaches based on reducing the amount of data via voxelization (Ahmed et al., 2019; Mescheder et al., 2018). However, there is no universal solution, and the amount of memory used and the complexity of preprocessing tend to make this process difficult (Wu et al., 2015).

In this study, the author demonstrates that accurate shape recognition can be achieved via learning based on neural networks using the solid angle as a quantitative feature for primitive shapes. In general, because one can express complex 3D shapes by performing Boolean operations on primitive shapes, constructive solid geometry representations (CSG) are widely used in the field of computer graphics (CG) and computer-aided design (CAD) (Fang, 2019). Therefore, a method for correctly learning and recognizing primitive shapes is required. By developing this research, it appears to be able to be utilized in methods such as CSG modeling to create complex shapes from primitive shapes (CORE CONCEPT TECHNOLOGIES INC., 2020; Hachiuma et al., 2017). Therefore, the application of this method may be effective in engineering fields that process geometry.

2. Related works

2.1. Solid angles

A solid angle, ω, comprising all the lines from a closed curve meeting at a vertex, is defined by the surface area of a sphere subtended by the lines and radius of that sphere, as shown in Fig. 1. The dimensionless unit of a solid angle is the steradian, with 4π steradians in a full sphere (Fig. 1). A Solid angle is the 3D equivalent of a 2D angle (Arecchi et al, 2007).

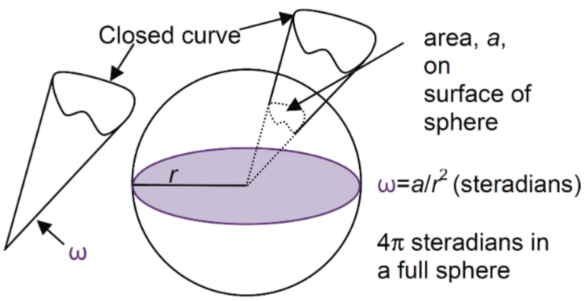


Fig. 1. Definition of a solid angle (Arecchi et al, 2007).

Because the surface area of a sphere is 4π, one can determine whether an arbitrary point belongs to the interior of a 3D figure using the solid angle (Fig. 2) (Arecchi et al, 2007; Kodama, 2018).

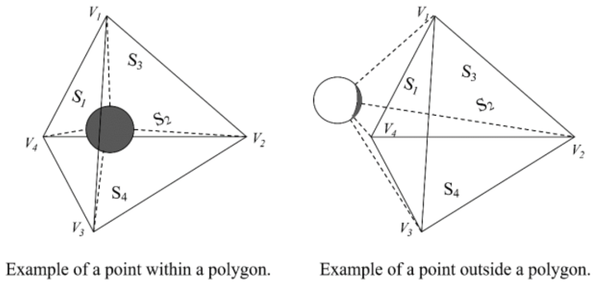


Fig. 2. Determining whether a point is within or outside a polygon based on the solid angle (Kodama, 2018).

Here, the curved surface S can distinguish between the front and the back. When point O is behind S, the point becomes $+P$; conversely, when point O is in front of S, the point becomes $-P$ (Fig. 3) (Feng, 2019; Kodama, 2018; Kodama, 2019).

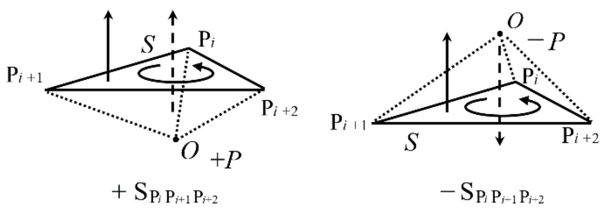
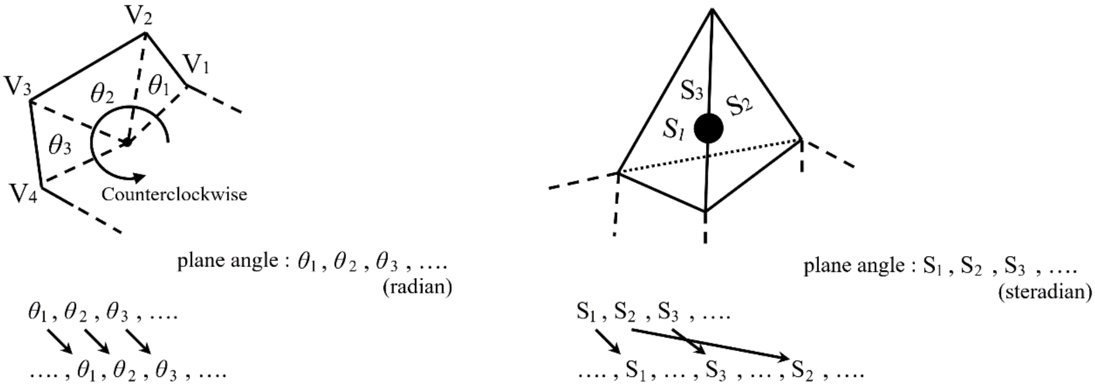


Fig. 3. Determining the obverse and reverse of a polygon (Kodama, 2018).

In a 3D shape, the input order of polygons is not uniquely determined (Research Institute for Computational Science, 2019), in contrast to a 2D shape; thus, the input can vary (Fig. 4).



The order of the obtained numbers does not change. The order of the obtained numbers is not uniquely determined.

Fig. 4. Input order of data in different dimensions.

This polygon method is widely used in CG, games, and movies because it can be easily and realistically expressed by pasting a texture on the surface. This method is also widely used in general software (Kato et al., 2019).

When a solid angle is used, one can accurately perform calculations even for complicated 3D shapes such as nonconvex shapes; however, as a disadvantage, the calculations are highly time-consuming due to the use of trigonometric functions (Kawakatsu et al, 1993; Nakayama et al, 1994). Therefore, when analyzing a complex shape using a solid angle, a specialized device may be needed, such as a General-Purpose computing on Graphics Processing Units (GPGPU), depending on the size of the object and the analysis range (Kodama, 2018). However, even if the object rotates, the solid angle of the set arbitrary point does not change. Therefore, a single value is obtained regardless of the posture of the object (Fig. 5).

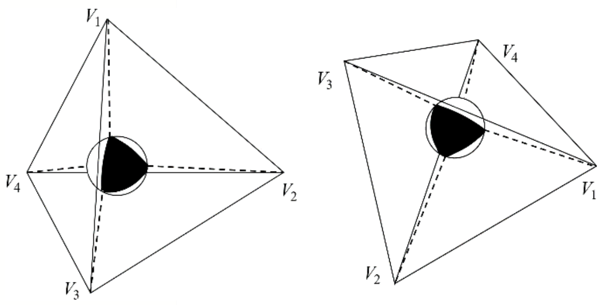


Fig. 5. The solid angle remains constant, even if the posture changes.

2.2 Learning and recognizing 3D shapes

2.2.1 Voxel-based solution

A voxel is an element of volume representing the value of a normal lattice unit in 3D space. Voxels are an extension of data pixels in 2D images and are often used for visualizing and analyzing medical and scientific data. Although voxelization has been used for a long time, its role has increased in recent years due to the continuous development of CPU and graphics hardware (Mileff and Dudra, 2019). Voxelization is frequently used because of its simplicity and ease in processing.

Using this method in combination with deep learning, a previous study proposed a technique for voxelizing a volume to 30×30×30 voxels (Fig. 6) (Wu et al., 2015). In addition, for 3D objects, various shapes can be used depending on the viewpoint. Researchers are seeking to improve identification systems by multitasking the learning process for object category identification and posture identification for rotation (Sedaghat et al., 2017).

Unfortunately, as the accuracy of a display depends on the size of each voxel or grid cell, the resolution is low and it is difficult to improve the recognition accuracy in voxelization methods (Le and Duan, 2018). If the voxels are small, a more detailed representation can be obtained; however, the memory increases in proportion to the third power, presenting a significant disadvantage (Shi et al., 2020).

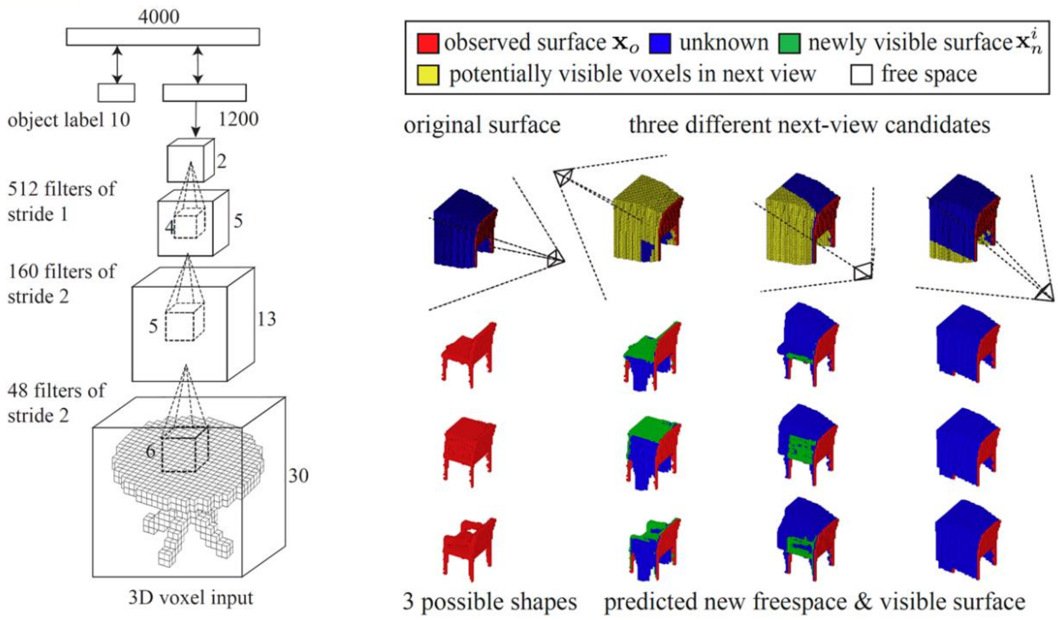


Fig. 6. Learning method based on voxelization (Wu et al., 2015).

2.2.2 Multiview solution

The multiview solution provides a method for object recognition and learning wherein 3D shapes are predicated based on 2D images obtained from various angle. Previous studies using this method have shown that the used data increase the categorical recognition accuracy by 8% compared with voxel-based methods (Su et al., 2015).

In the multiview solution, a large number of virtual cameras facing the center of gravity are arranged around the axis to create a number of 2D images and 3D shapes are compared based on these images. Subsequently, images obtained using numerous cameras around the axis are individually inputted to a convolutional neural network (CNN), and the obtained feature map is integrated using a pooling layer (view-pooling) to obtain invariance with respect to rotation around the axis (Fig. 7) (Kanezaki et al, 2018; Su et al., 2015). In recent studies, attention fusion has been conducted based on multiview images and point clouds to improve accuracy (You et al., 2018).

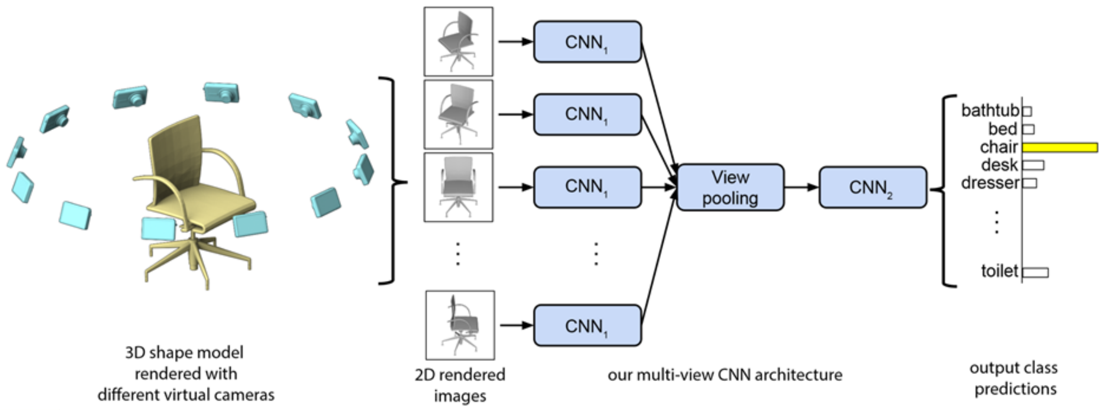


Fig. 7. Learning method based on the multiview solution (Su et al., 2015).

2.2.3 Spin-image solution

Johnson and Hebert generated a spin image by moving surrounding vertices on a cylindrical surface centered on the normal vector of a vertex (Johnson, 1997; Johnson and Hebert, 1999). They proposed a method of correspondence based on a search for similar images using the spin image compressed via primary component analysis (Fig. 8).

Currently, researchers are seeking to collate an input distance image by projecting the positional relationship with respect to peripheral points in 2D based on the normal vector at the target point of the model and acquiring features that do not depend on the posture (Deng et al, 2018).

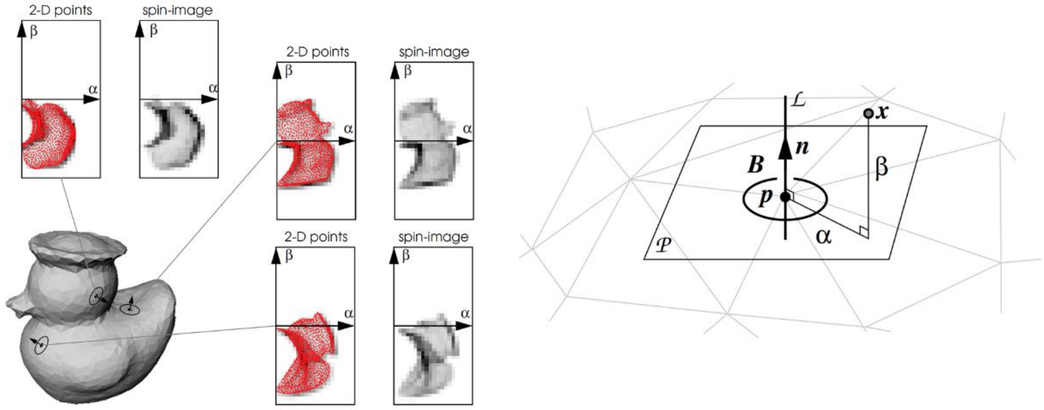


Fig. 8. Acquisition of features based on spin images (Johnson, 1997).

2.2.4 Learning and recognition through deep learning

For learning and recognition, deep learning methods have been studied based on quantitative features obtained using the voxel-based solution, multiview solution, and spin-image solution, as previously presented (Ahmed et al., 2019; Lui et al., 2019; Varma et al., 2020). However, unlike 2D shapes, 3D shapes must be observed from various directions, and one must determine whether the object remains the same when rotated (AI-SCHOLAR, 2018). In addition, the range of notable shapes may be broad, or a detailed range may be required. Therefore, it is difficult to apply methods based on CNNs (Fig. 9) (Fang et al., 2015; Jurafsky and Martin, 2020).

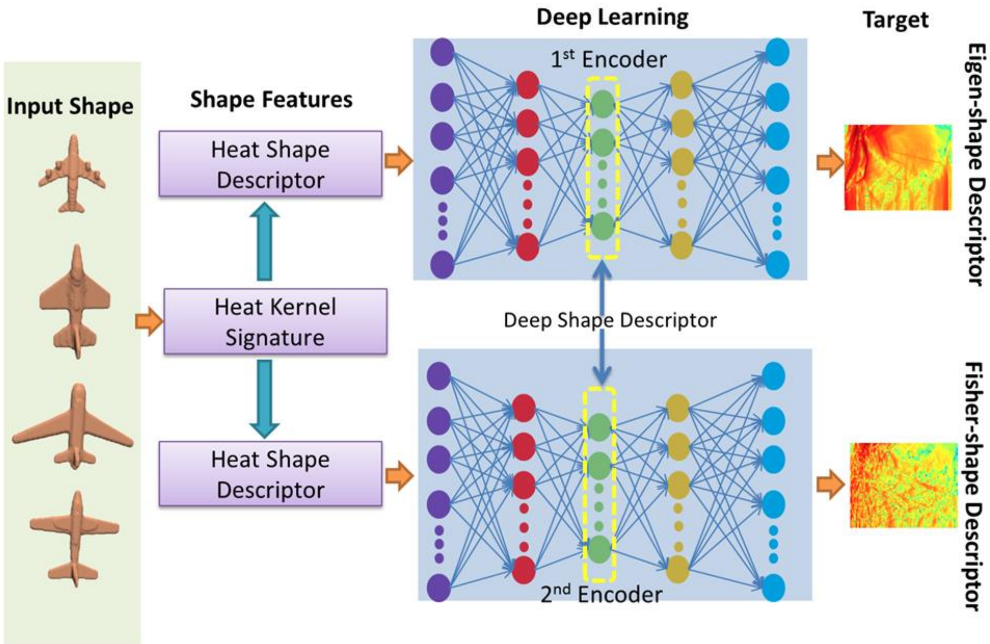


Fig. 9. Learning method for 3D figures based on deep learning (Fang et al., 2015).

2.3 Neural network

A neural network is a combination of artificial neuros, i.e., computational elements that model nerve cells. Figure 10 shows the composition of a single artificial neuron. The artificial neuron receives multiple input signals, performs appropriate calculations, and then provides an output signal.

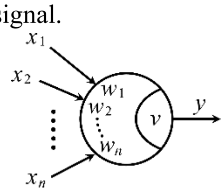


Fig. 10. Composition of an artificial neuron (Okada, 2016).

For a multi-input/single-output nonlinear element, as shown in Fig. 10, the neuron receives a multidimensional input $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ and outputs y.

In general, the processing of Fig. 10 can be described via Eq. (1).

$$y = f\left(\sum_{i=1}^{n}(w_i x_i - \theta)\right) \qquad \text{(Eq. 1)}$$

where x_i is the i-th element of x and w_i is the coupling weight corresponding to the i-th input, a parameter that represents synaptic signaling. Furthermore, θ is a threshold value. Function f is an activation function that is often expressed as a sigmoid function, such as that given in Eq. (2) (Jurafsky and Martin, 2020).

$$f(u) = \frac{1}{1 + e^{-u}} \qquad \text{(Eq. 2)}$$

In general, learning is performed by constructing a network that combines numerous artificial neurons and then connecting the outputs and inputs in order. For example, a neural network that receives two input signals and provides one output signal can be built as shown in Fig. 11 (Odaka, 2016). For this type of structure, the configuration in which the signal propagates from input to output is called a feedforward network or a layered network (Odaka, 2016; Jurafsky and Martin, 2020).

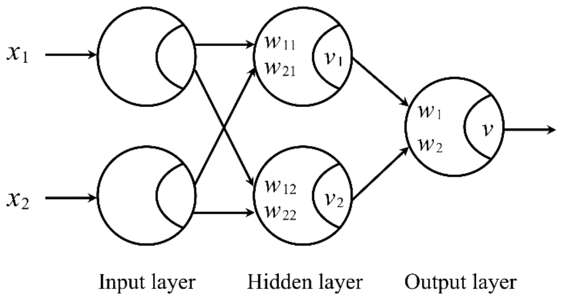


Fig. 11. Example of a feedforward network configuration (Okada, 2016).

Researchers have applied hierarchical neural networks to various tasks such as character recognition, phonological recognition, and signal processing. It has been shown that tasks such as handwritten numeral recognition can achieve almost the same performance as that obtained by manual classification. For example, previous studies have shown that a network with a total of nine layers can be applied to raw data encountered in face recognition, image recognition, handwritten characters, and word vectors in documents and can extract the structure latent in the data as a feature (Oord et al., 2016; Hinton and Salakhutdinov, 2006; Kalchbrenner et al., 2016; Zhang et al, 2016). In addition, for successful learning via neural networks with a large number of layers, researchers have developed a method for reducing the degree of freedom of coupling weights and facilitating learning by creating a task-specific coupling structure in advance (Fukushima, 2013). This approach corresponds to CNN and is often used for image recognition. CNN is a very effective method, but on the other hand, it has been pointed out that it is difficult to apply directly to numerical data such as CSV files (Takahashi et al., 2018).

For learning in multilayer networks, in principle, it has been proven that arbitrary input/output relationships can be achieved if there is one intermediate layer with a sufficiently large number of neurons (Cybenko, 1989; Funahashi, 1989).

3. Proposed method

In the proposed method, the author creates primitive shapes for multiple 3D shapes and calculate each solid angle. Thereafter, machine learning is performed using a neural network based on the obtained solid angle. It should be noted that this research assumes a preliminary step to develop into a Deep Neural Network (DNN), and for this purpose, a layered neural network has been built and verified. In other words, the effectiveness is verified by the basic configuration as based on Figure 11.

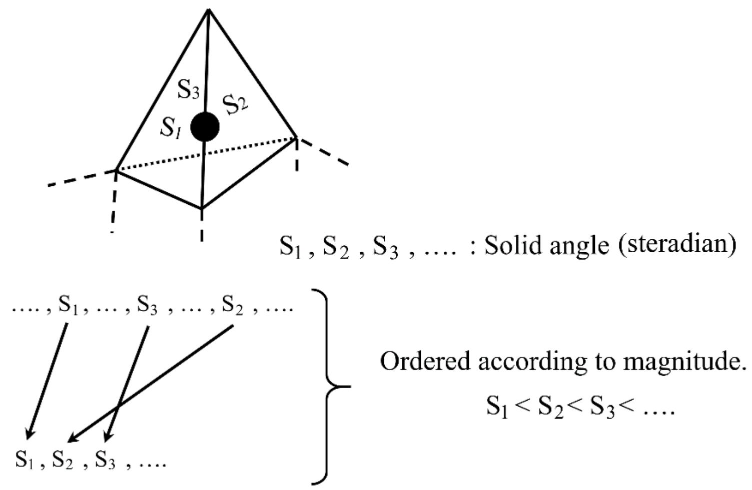


Fig. 12. Sorting of solid angles by magnitude (steradians).

As mentioned earlier, the order in which data are entered is generally not uniquely determined for 3D shapes, in contrast to 2D shapes (Fig. 4). Therefore, the author creates triangular polygons, and the solid angles of each polygon are sorted according to their magnitude, followed by the learning process (Fig. 12). In addition, in this research, the solid angle calculated for each triangle polygon centered at an arbitrary single point was used.

As an example, for the triangular pyramid shown in Fig. 13, the solid angle centered at the origin has four values, and a unique numerical sequence is obtained by applying the sorting method (Fig. 14).

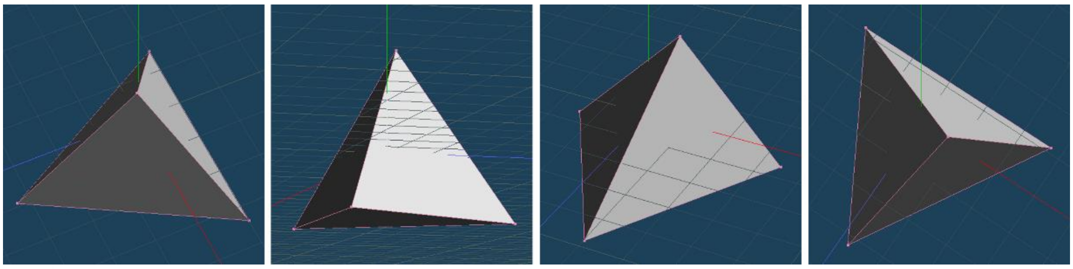


Fig. 13. Rotations of a triangular pyramid.

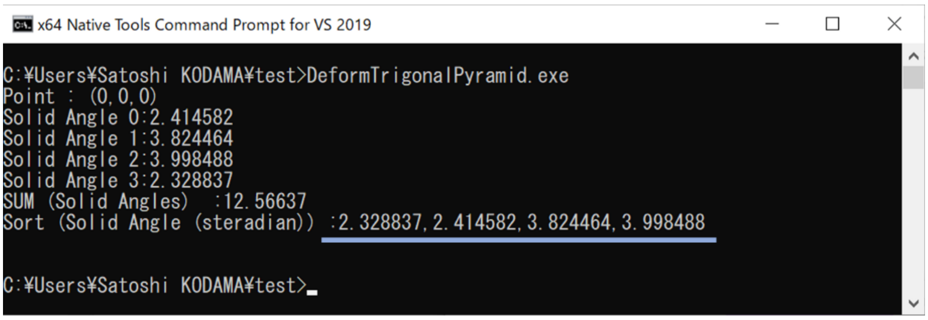


Fig. 14. Example of sorting the obtained solid angles (steradians).

Subsequently, the numerical sequence is learned by the neural network. The structure of the neural network used for learning has n inputs, single hidden layer with n and one output (Fig. 15). Here, the number of inputs n corresponds to the number of triangular polygons and x is the value of the sorted solid angle. In this configuration, a sigmoid function (Eq. 2) was used as an activation function, and the initial learning rate was set at 0.5.

Verification data were created using the same method.

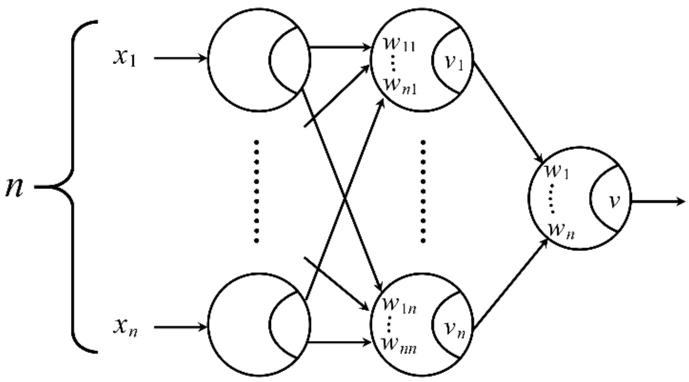


Fig. 15. Structure of neural networks used in this experiment.

4. Experiments and evaluation

The solid angles for cubes, rectangular shapes, cones, cylinders, and shapes were calculated for the learning process, which was implemented via neural networks. The effectiveness of the solid angle learning method was verified by creating a shape for verification and comparing this shape with the learning result.

As this study aimed to confirm the effectiveness of learning with solid angles as a feature, only the solid angle centered on the coordinate origin was used. The structure of the neural network used for learning changes the number of input n according to the number of triangular polygons, as illustrated in Fig. 15. Furthermore, the parameters of each cell were automatically changed by entering training data. After that, it was judged whether the result of entering the test data was correct for the parameters obtained by the training data. As mentioned above, since this is an initial verification of the feature, note that the experiment was conducted with a focus on dividing the two primitive shapes that are often used for CSG trees.

The specifications of the used equipment are shown in Table 1. No specialized devices were used in this experiment.

Table 1. Specifications of the equipment used in this experiment.

Specification	Value
OS	Windows 10 Professional (1909)
Compiler	Visual C# Compiler version 3.0.19.21801 (Visual Studio 2019)
CPU	Intel Core i7-8565U (1.8 GHz)
Memory	16 GB (DDR3)

4.1 Cube and rectangular shape (triangular polygons = 12)

The solid angle was calculated for the cubes and rectangular shapes shown in Figs. 16 and 17, respectively. Then, the solid angles of the obtained triangular polygons were rearranged in ascending order and used as learning data. In this case, because the solid angle was calculated from the coordinate origin, the solid angles are the same for each cube (Fig. 16).

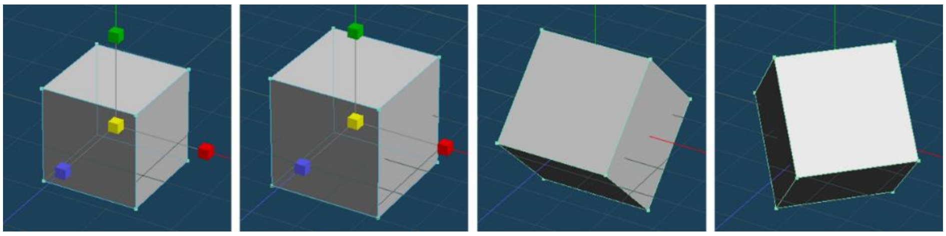


Fig. 16. Cubes utilized for learning.

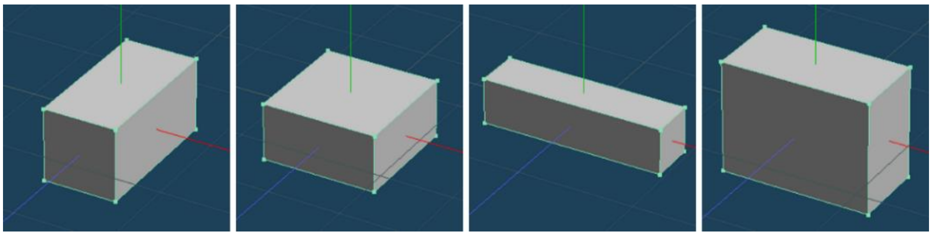


Fig. 17. Rectangular shapes utilized for learning.

For verification data, the 3D images shown in Fig. 18 were used. The verification data were processed in the same manner as the learning data, and the author verified that the shapes were correctly identified. The recognition rates based on the learning results are shown in Table 2.

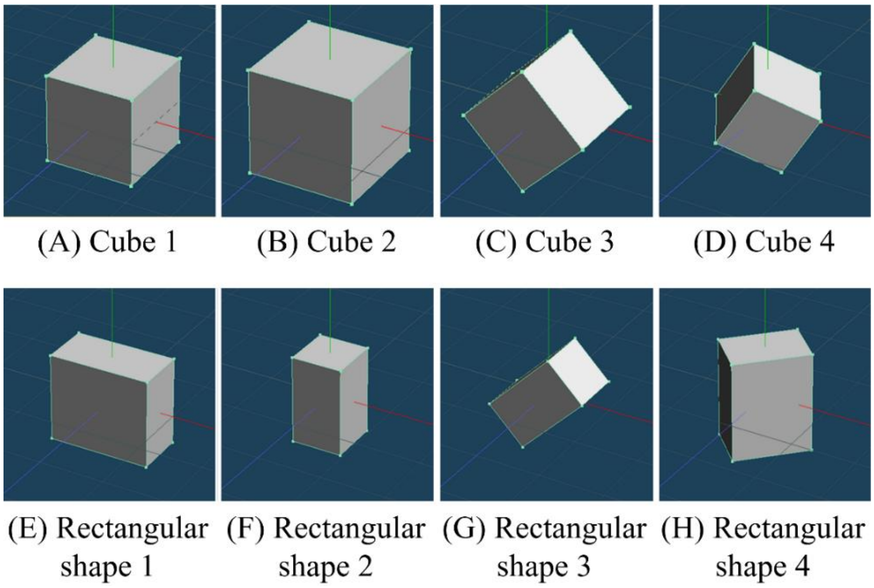


Fig. 18. Solid shapes for validation.

Table 2. Recognition rates of 3D shapes for verification.

3D shape	Correct identification rates (%)
(A) Cube 1	99.290124
(B) Cube 2	99.290124
(C) Cube 3	99.290124
(D) Cube 4	99.290124
(E) Rectangular shape 1	99.609371
(F) Rectangular shape 2	99.187627
(G) Rectangular shape 3	99.187628
(H) Rectangular shape 4	99.187628

4.2 Cone and rectangular shape (triangular polygons = 12)

The solid angle was calculated for the cones and rectangular shapes shown in Figs. 19 and 20, respectively. Then, the solid angles of the obtained triangular polygons were rearranged in ascending order and used as learning data.

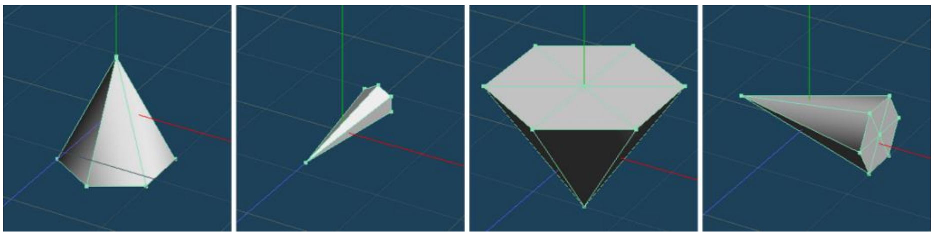


Fig. 19. Cones utilized for learning.

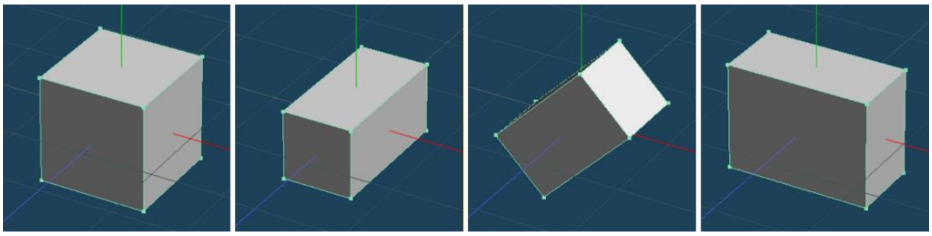
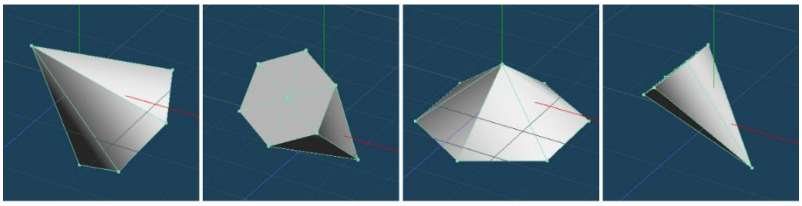
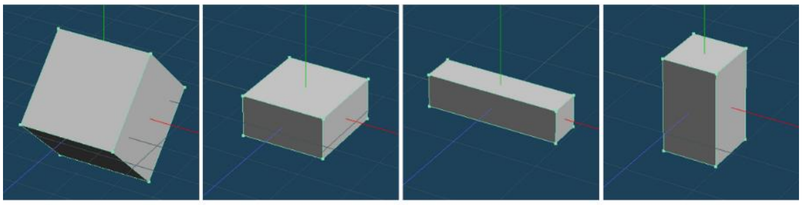


Fig. 20. Rectangular shapes utilized for learning.

For verification data, the 3D images shown in Fig. 21 were used. The verification data were processed in the same manner as the learning data, and the author verified that the shapes were correctly identified. The recognition rates based on the learning results are shown in Table 3.



(A) Cone 1 (B) Cone 2 (C) Cone 3 (D) Cone 4



(E) Rectangular shape 1 (F) Rectangular shape 2 (G) Rectangular shape 3 (H) Rectangular shape 4

Fig. 21. Solid shapes for validation.

Table 3. Recognition rates of 3D shapes for verification.

3D shape	Correct identification rates (%)
(A) Cone 1	99.812166
(B) Cone 2	99.900417
(C) Cone 3	99.324063
(D) Cone 4	99.885613
(E) Rectangular shape 1	99.949864
(F) Rectangular shape 2	98.970878
(G) Rectangular shape 3	97.931441
(H) Rectangular shape 4	99.792057

4.3 Cone and cylinder (triangular polygons = 24)

The solid angle was calculated for the cones and cylinders shown in Figs. 22 and 23, respectively. Then, the solid angles of the obtained triangular polygons were rearranged in ascending order and used as learning data.

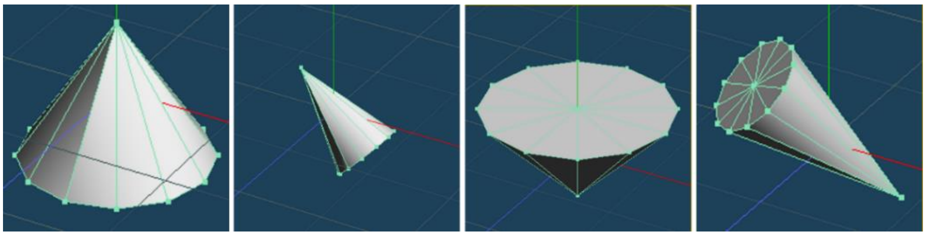


Fig. 22. Cones utilized for learning.

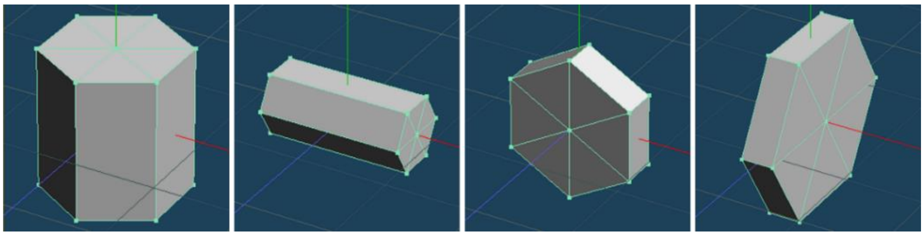


Fig. 23. Cylinders utilized for learning.

For verification data, the 3D images shown in Fig. 24 were used. The verification data were processed in the same manner as the learning data, and the author verified that the shapes were correctly identified. The recognition rates based on the learning results are shown in Table 4.

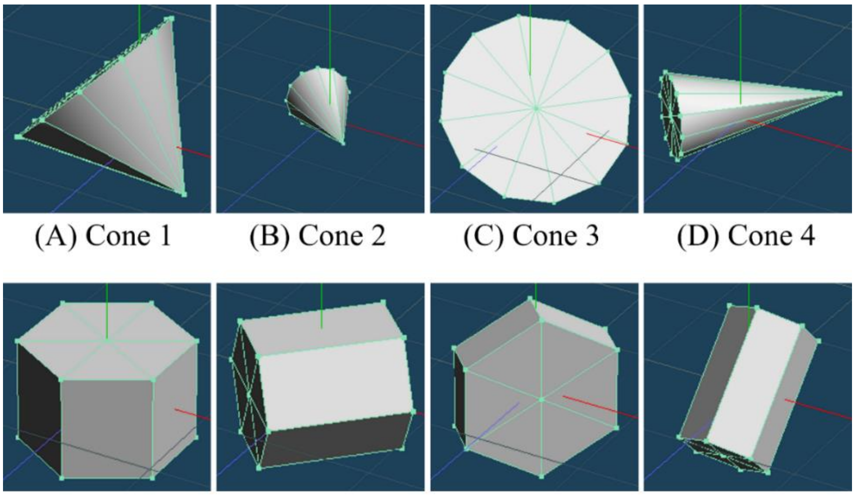


Fig. 24. Solid shapes for validation.

Table 4. Recognition rates of 3D shapes for verification.

3D shape	Correct identification rates (%)
(A) Cone 1	98.864965
(B) Cone 2	99.549863
(C) Cone 3	99.995699
(D) Cone 4	99.988801
(E) Cylinder 1	99.925257
(F) Cylinder 2	99.925258
(G) Cylinder 3	98.021626
(H) Cylinder 4	99.999792

4.4 Cylinder and sphere shapes (triangular polygons = 40)

The solid angle was calculated for the cylinders and spheres shown in Figs. 25 and 26, respectively. Then, the solid angles of the obtained triangular polygons were rearranged in ascending order and used as learning data.

For verification data, the 3D images shown is Fig. 27 were used.

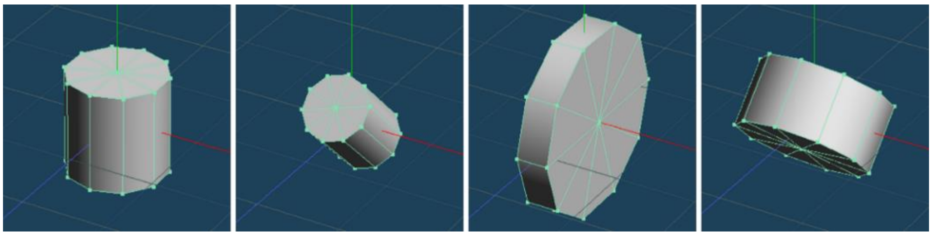


Fig. 25. Cylinders utilized for learning.

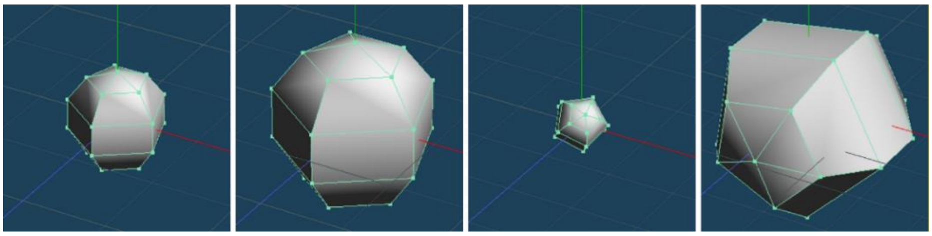


Fig. 26. Spheres utilized for learning.

The verification data were processed in the same manner as the learning data, and the author verified that the shapes were correctly identified. The recognition rates based on the learning results are shown in Table 5.

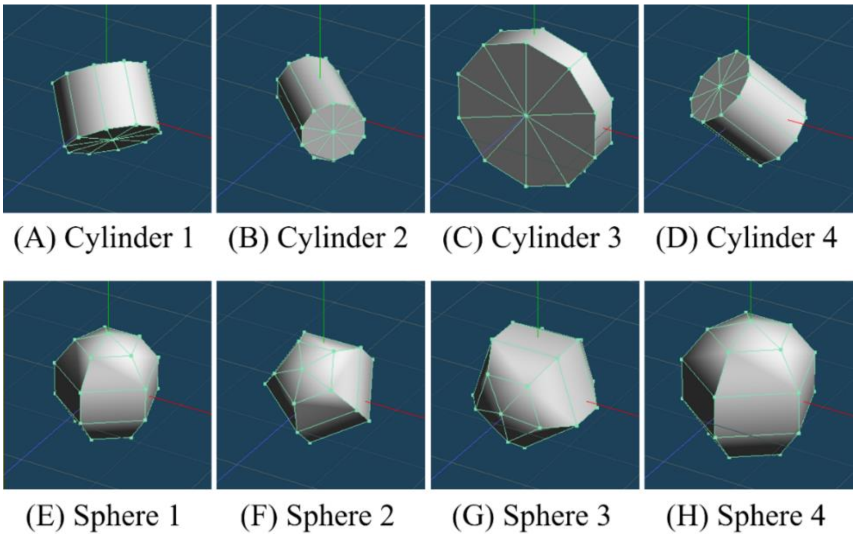


Fig. 27. Solid shapes for validation.

Table 5. Recognition rates of 3D shapes for verification.

3D shape	Correct identification rates (%)
(A) Cylinder 1	97.614127
(B) Cylinder 2	99.739156
(C) Cylinder 3	99.608418
(D) Cylinder 4	98.213890
(E) Sphere 1	98.472752
(F) Sphere 2	99.986135
(G) Sphere 3	99.999984
(H) Sphere 4	98.472755

5. Conclusion

Using the solid angle as a quantitative feature of 3D shapes, it was possible to identify a given shape with a high recognition rate. Therefore, learning based on a neural network using a solid angle as a quantitative feature is an effective means for recognizing 3D shapes.

In contrast to conventional methods, this method utilizes solid angles, and one can easily extract feature quantities without requiring a setup with virtual cameras or positional relationships for each point, as is needed for conventional learning methods. In addition, although the operations for the solid angle are time-consuming, the processing is relatively simple.

6. Future work

This study primarily focused on confirming the effectiveness of learning using solid angles; thus, primitive 3D shapes were evaluated. Furthermore, because this learning employed a small-scale neural network, experiments were conducted on polygons with a relatively small amount of data. Therefore, in this experiment, it was possible to verify without using special devices, but when expanding it in the future, it is necessary to consider the use of device such as GPGPU.

In general, complex 3D shapes require a large amount of data. In addition, as there are various methods for expressing 3D shapes, one must be able to recognize a figure using various data representations.

In the future, evaluations of more complex shapes will be required. As the amount of data grows with increasing complexity, further learning and evaluation using deep learning or similar methods must be performed. In addition, it is necessary to consider its application to CNNs. Furthermore, the use of solid angles is time-consuming for complex 3D shapes; therefore, the overall processing time should be assessed.

By advancing this research, it is thought that shape recognition from a point cloud will be possible by Light Detection and Ranging (LiDAR), Sound Navigation and Ranging (SONAR) 3D mapping, etc., and its application in the industrial field is expected to expand.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP19K14599.

References

Ahmed, E., Saint, A., Shabayek, A. E. R., Cherenkova, K., Das, R., Gusev, G., Aouada, D., & Ottersten, B. (2019). A survey on deep learning advances on different 3D data representations, arXiv preprint arXiv:1808.01462.

AI-SCHOLAR. (2018). Available on line at: https://ai-scholar.tech/articles/treatise/deeplearning-3d-36 (accessed on November 12, 2020)

Arecchi, V. A., Messadi, T., & Koshel, J. R. (2007). *Field Guide to Illumination (Field Guide Series)*, Society of Photo Optical.

Cohen, T. S., Geiger, M., Köehler, J., & Welling, M. (2018). Spherical CNNs, Proceedings of the 6th International Conference on Learning Representations (ICLR), arXiv preprint arXiv: 1801.10130.

CORE CONCEPT TECHNOLOGIES INC., Available on line at: https://recruit.cct-inc.co.jp/tecblog/com-geometry/3d-cad-fr/ (accessed on November 22, 2020)

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals, and Systems, 2, 303-314. https://doi.org/10.1007/bf02551274.

Deng, H., Birdal, T., & Ilic, S. (2018). PPFNet: Global context aware local features for robust 3D point matching, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, arXiv preprint arXiv:1802.02669.

Eleliemy, A., Mohammed, A., & Ciorba, F. M. (2017). Efficient generation of parallel spin-images using dynamic loop scheduling, IEEE 19th International Conference on High Performance Computing and Communications Workshops (HPCCWS). https://doi.org/10.1109/hpccws.2017.00012.

Fang, H. (2019). Geometric modeling of man-made objects at different level of details. Université Côte d'Azur. Available on line at: https://tel.archives-ouvertes.fr/tel-02406834/file/2019AZUR4002.pdf (accessed on December 23, 2020).

Fang, Y., Xie, J., Dai, G., Wang, M., Zhu, F., Xu, T., & Wong, E. (2015). 3D deep shape descriptor, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), https://doi.org/10.1109/CVPR.2015.7298845.

Feng, Y., Feng, Y., You, H., Zhao, X., & Gao, Y. (2019). MeshNet: Mesh neural network for 3D shape representation, Proceedings of the AAAI Conference on Artificial Intelligence, 33(01). 33(1). https://doi.org/10.1609/aaai.v33i01.33018279.

Fukushima, K. (2013). Artificial vision by multi-layered neural networks: Neocognitron and its advances, (2013)37. 103-119. https://doi.org/10.1016/j.neunet.2012.09.016.

Funahashi, K. (1989). On the approximate realization of continuous mappings by neural networks, Neural Networks, 2(3), 183-192. https://doi.org/10.1016/0893-6080(89)90003-8.

Hachiuma, R., Ozasa, Y., & Saito, H. (2017). Primitive Shape Recognition via Superquadric representation using large margin nearest neighbor classifier. In Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017), 5, 325 - 332.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks, Science, 313(5786). 504-507. https://doi.org/10.1126/science.1127647.

Johnson, A. & Hebert, M. (1999). Using Spin Images for Efficient Object recognition in cluttered 3D scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(5), 433 - 449. https://doi.org/10.1109/34.765655.

Johnson, A. E. (1997). *Spin-Images: A representation for 3-D surface matching*, Ph. D Thesis, CMU-RI-TR-97-47, Robotics Institute, Carnegie Mellon University, Available on line at: https://www.ri.cmu.edu/pub_files/pub2/johnson_andrew_1997_3/johnson_andrew_1997_3.pdf (accessed on October 27, 2020)

Jurafsky, D., & Martin, J. H. (2020). Speech and language processing (Third edition draft), Available on line at: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf. (accessed on December 28, 2020).

Kalchbrenner, N., Espeholt, L., Simonyan, K., Oord, A., Graves, A., & Kavukcuoglu, K. (2016). Neural machine translation in linear time, arXiv preprint arXiv: 1610.10099.

Kanezaki, A., Matsushita, Y., & Nishida, Y. (2018). RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints, Computer Vision and Pattern Recognition, arXiv preprint arXiv: 1603.06208.

Kato, Y., Tanaka, S., Kanamori, Y., & Mitani, J. (2019). Single-View Modeling of Layered Origami with Plausible Outer Shape, Computer Graphics Forum, 38(7), 629 – 640, https://doi.org/10.1111/cgf.13866.

Kawakatsu, D., Nakayama, A., Kobori, K., & Kutsuwa, T. (1993). A method of selecting an object in virtual reality, IPSJ SIG on CGVI. 110, 25 - 32. (in Japanese)

Kodama, S. (2018). Effectiveness of inside/outside determination in relation to 3d non-convex shapes using cuda, The Imaging Science Journal, 66(7), 409-418. https://doi.org/10.1080/13682199.2018.1497251.

Kodama, S., Nakagawa, R., & Ozeki, Y. (2019). Rapid determination of three-dimensional convex shapes by dispersion processing using java RMI, International Journal of Computer Science and Engineering, 6 (11), 18 – 27. https://doi.org/10.14445/23488387/ijcse-v6i11p105.

Le, T., & Duan, Y. (2018). PointGrid: A deep network for 3D shape understanding, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9204 - 9214, https://doi.org/10.1109/cvpr.2018.00959.

Lui, M., Yao, F., Choi, C., Sinha, A., & Ramni, K. (2019). Deep learning 3D shapes using alt-az anisotropic 2-sphere convolution, Proceedings of the 7th International Conference on Learning Representations.

Mescheder, L., Oechsle, M., Niemeyer M., Nowozin, S., & Geiger, A. (2018). Occupancy networks: Learning 3D reconstruction in function space, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 4455-4465. https://doi.org/10.1109/CVPR.2019.00459.

Mileff, P., & Dudra, J. (2019). Simplified voxel based visualization. Production Systems and Information Engineering, 8, 5 - 18. https://doi.org/10.32968/psaie.2019.001.

Nakayama, A., Kawakatsu, K., Kobori, K., & Kutsuwa T. (1994). A checking method for a point inside a polyhedron in grasping an object of VR, Proceedings of the 48th National Convention of IPSJ, 297 - 298. (in Japanese)

Odaka, T. (2016). *Machine learning and deep learning*, Tōkyō: Ōmusha. (in Japanese)

Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). WaveNet: A generative model for raw audio, arXiv preprint arXiv: 1609.03499.

Research Institute for Computational Science. (2019). Available on line at: https://www.ricos.co.jp/tech-0101_ml_fem_geometry/ (accessed on October 25, 2020).

Sedaghat, N., Zolfaghari, M., Amiri, E., & Brox, T. (2017). Orientation-boosted Voxel Nets for 3D Object Recognition, Procedings of the British Machine Vision Conference 2017, https://doi.org/10.5244/c.31.97.

Shi, Z., Lin, Y., & Li, H. (2020). Extraction of urban power lines and potential hazard analysis from mobile laser scanning point clouds, International Journal of Remote Sensing, 41(9), https://doi.org/10.1080/01431161.2019.1701726.

Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). Multi-view convolutional neural networks for 3D shape recognition, 2015 IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/iccv.2015.114.

Takahashi, K., Numajiri, T., Sogabe, M., Sakamoto, K., Yamaguchi, K., Yokogawa, S., & Sogabe, T. (2018). Development of general-purpose CNN deep layer learning method using feature graph, The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 4Pin1-01, 1-4. (in Japanese)

Varma, M. J., Kx, V. K., Thorannath, K. V., Harshavardhan, C. A, & Ahmed, M. R. (2020). 3D reconstruction of 2d images using deep learning on the nvidia jetson nano, International Journal of Advanced Science and Technology, 29(10s), 7681-7686. Available on line at: http://sersc.org/journals/index.php/IJAST/article/view/24090. (accessed on November 25, 2020).

Wang, Y., Pan, G., Wu, Z., & Han, S. (2004). Sphere-Spin-Image: A Viewpoint-Invariant surface representation for 3D face recognition, Computational Science, ICCS 2004 Lecture Notes in Computer Science, 427-434, https://doi.org/10.1007/978-3-540-24687-9_54.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., & Xiao, J. (2015). 3D ShapeNets: A deep representation for volumetric shapes, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), https://doi.org/10.1109/cvpr.2015.7298801.

You, H., Feng, Y., Ji, R., & Gao, Y. (2018). PVNet: A joint convolutional network of point cloud and multi-view for 3D shape recognition, MM '18: Proceedings of the 26th ACM international conference on Multimedia, 1310 - 1318.