Learning and Recognition of Primitive Three-dimensional Shapes using Solid Angles

Satoshi Kodama[#1], Kota Endo[*2]

[#] *Research Institute for Science and Technology, Tokyo University of Science*
2641 Yamazaki, Noda, Chiba Prefecture, Japan

[*] *Department of Information Sciences, Graduate School of Science and Technology, Tokyo University of Science*
2641 Yamazaki, Noda, Chiba Prefecture, Japan

Abstract — *Unlike two-dimensional shapes, recognition of three-dimensional shapes has various backgrounds depending on the viewpoint and rotation of the object; thus, shape recognition using machine learning techniques is very difficult. Generally, the black box nature of deep learning is problematic from an accountability perspective.*

Therefore, in this paper, we propose a method to evaluate the similarity for learning by focusing on polygons that make up a three-dimensional shape and the solid angle for them. The value and pattern of the three-dimensional angle differ for each object shape; thus, by classifying based on the results of the proposed method, it is possible to perform learning and shape recognition.

In this study, we verified the effectiveness of learning using three-dimensional angles and shape recognition by creating multiple primitive shapes as samples. In addition, we verified for three-dimensional shape deformed part of the three-dimensional primitive shape. As a result, using the solid angle, it was confirmed that can be effectively determined for three-dimensional primitive shapes.

Keywords — *Shape recognition, Shape learning, Pattern recognition, Shape Registration, Solid angle.*

I. INTRODUCTION

Three-dimensional shapes differ significantly from two-dimensional shapes, and three-dimensional shapes must be observed from various directions to determine if objects are the same when rotated [1, 2]. Therefore, unlike two-dimensional shapes, it is difficult because appropriate processing must be performed each time depending on the rotation and viewing angle of the object. Generally, the recognition and learning method of three-dimensional shape, a method using a learning method and normal vector by images from various angles, also, techniques such as dealing with by reducing the number of data by voxelization has been studied so far [3, 4]. However, there is no universal solution because the amount of data becomes enormous; thus, it is difficult to process. Thus, it is difficult to recognize and learn a three-dimensional shape [1, 2, 4]. In recent years, black boxes for learning methods have been a problem [5, 6], i.e., accountability is required; thus, a method to extract feature points for shape learning is required [5, 7, 8].

II. RELATED WORK

A. *Three-dimensional object recognition*

Generally, three-dimensional shape analysis uses countermeasures based on a voxel base, normal vectors, and multi-view. In addition, deep learning methods have been studied for learning and recognition [4, 9, 10].

a) *Voxel-based solution*

A voxel is a generalization of pixel to 3D, since only extended the 2D image (image), is the most commonly used expression from its simplicity [10]. A previous study proposed a deep learning method after converting it to 30 × 30 × 30 [11]. In the case of a three-dimensional shape, depending on the viewpoint, but a variety of shapes, for rotation, by performing multitasking learning of category identification and posture identification of the object, research to improve the identification system has also been carried out [12]. The accuracy of the display is determined by the size of each voxel or grid cell. Generally, it becomes low resolution; thus, various problems, such as improving recognition accuracy, are known [10]. In particular, with voxelization, the spatial characteristics of point cloud data change and uneven expression of the object is lost [13].

b) Spin Image solution

This method is based on the normal vector at the point of interest in the object model, and the relative positional relationship with the surrounding points is projected onto a two-dimensional projection plane. With this method, collation independent of the viewpoint is possible. However, all points of the object model or points used for matching are selected randomly; thus, a point with a similar local shape may be selected, which can result in erroneous determination. Generally, the number of points used for matching is large; thus, processing cost has become problematic [14].

c) Multi-view solution

Based on a two-dimensional image obtained from various angles and by predicting the three-dimensional shape, it is an object recognition method. The category recognition accuracy of this method is increased by 8% compared to the voxel-based ShapeNet [15]. Specifically, a large number of virtual cameras directed toward the center of gravity are arranged around the vector axis of the 3D model to create a multiple 2D rendering images, and the 3D shapes are compared based on these images. Then, by inputting images from a large number of cameras around the axis individually into a CNN, and integrating the resulting feature map in a pooling layer called View-Pooling, we can obtain invariant to rotation around the axis. In recent studies, attention fusion has been performed using multi-view images and point clouds to improve accuracy [16].

B. Solid angle

A solid angle corresponds to an angle in a two-dimensional shape, and the solid angle is the surface area of a sphere that exists in three dimensions (Fig. 1) [17]. Generally, from this property, it is possible to determine whether an arbitrary point belongs to the inside of a solid figure (Fig. 2) [17 - 19].

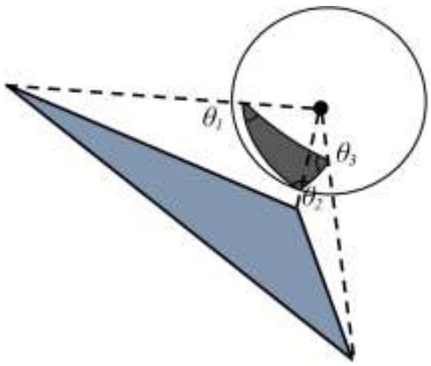

Fig. 1. When projecting a sphere from the polygon

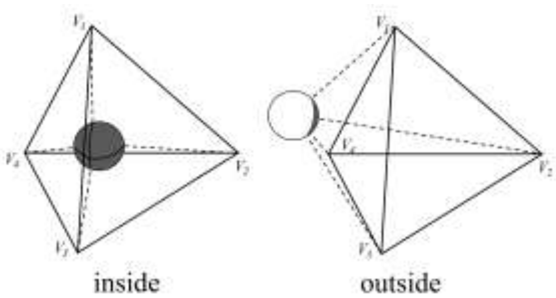

Fig. 2. Example of inside / outside a polygon

The solid angle can be obtained using Equation (1).

$$S = (\theta_1 + \theta_2 + \theta_3 - \pi) \times r^2 \qquad (1)$$

Here, r is the radius of sphere (Fig. 1) [17 - 19].

In the inside / outside determination method using an angle, a two-dimensional shape can be determined using the clockwise direction as a reference. For a three-dimensional shape, determination is performed according to the polygon input direction (Figs. 3 and 4) [18 - 21]. This input method is a general polygon description method; thus, when modeling software is used, determination can be made without being conscious [18, 20].

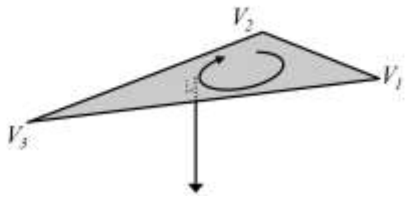

Fig. 3. A triangle with its surface normal (winding order for a triangle)

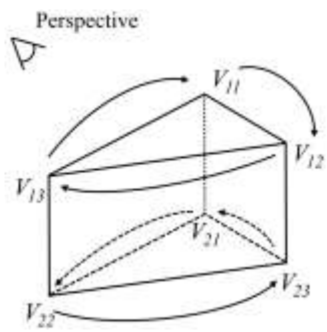

Fig. 4. Example of data input in relation to object

With a determination method that uses a solid angle, it is possible to accurately determine complicated three-dimensional shapes, including non-convex shapes. However, many trigonometric functions are required; thus, determination takes a very long time [18, 19].

III. PROPOSED METHOD

A. Create solid angle-based database

We propose to create a database of three-dimensional shapes to be learned using solid angle. First, a solid angle is calculated for each vertex (point) comprising the primitive three-dimensional shape. Therefore, the points used to calculate the solid angle are the points entered to form the polygon (Fig. 5). For example, four points are registered in the case of a triangular pyramid, and eight points are registered in the case of a hexahedron (Fig. 6). Then, learning is performed by registering the value in the database. Some of the values to be registered in the database are shown in Fig. 7. When learning, it is assumed that a set of labels and model data is used.

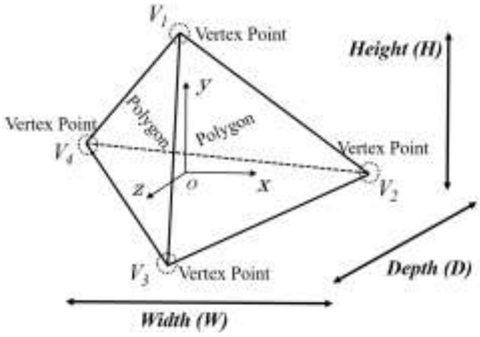

Fig. 5. Composition of 3D objects

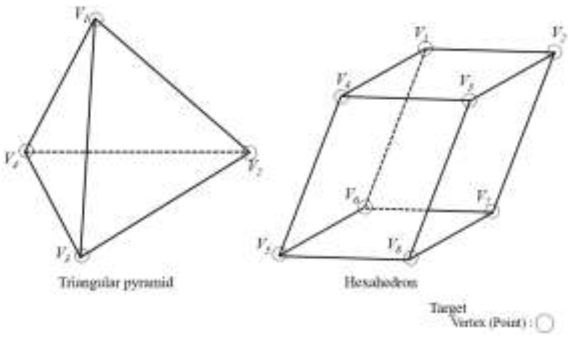

Fig. 6. Example of a target point

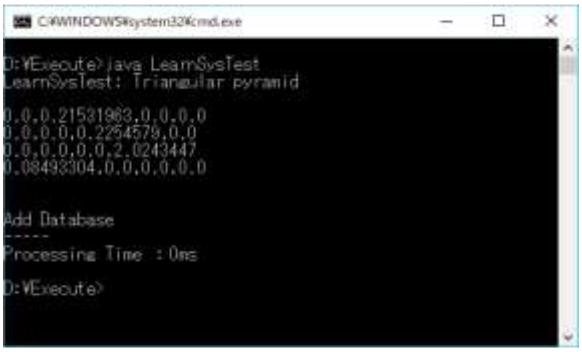

Fig. 7. Add shapes to the database

B. Dynamic search for similarity

The similarity regarding the learning data is determined by dynamically changing the threshold value based on the number of corresponding vertices and the solid angle value in the database (Table 1 and Fig. 8). When outputting the results, it returns the label of the data with the lowest error at each point.

Table. 1. Relationship between model and solid angles

	Vertex (Triangular pyramid)			
Polygons	0	0.21531963	0	0
	0	0	0.22545790	0
	0	0	0	2.02434470
	0.08493304	0	0	0

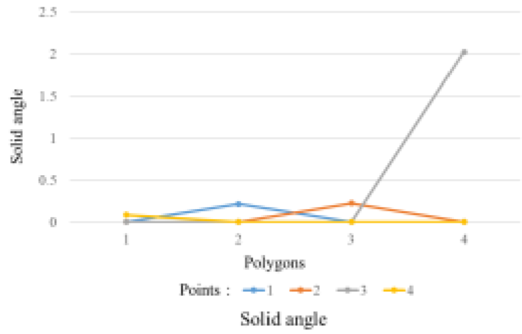

Fig. 8. Examples of feature points extraction

IV. EXPERIMENT

A database was created using the solid figure in Figs. 9 to 13 in a three-dimensional space of -50<x<50, -50<y<50, -50<z<50. Then, the degree of similarity with the created database (Figs. 9 to 13) was calculated for Figs. 14 to 18 as a new solid figure (with scale change and rotation). The details of each figure are shown in Tables 2 and 3. Figures 19 to 23 show the solid angle values in Figs. 9 to 13 used for learning. In addition, the solid angle values in Figs. 14 to 18 are shown in Figs. 24 to 28.

We also experimented with some deformed figures. In this experiment, Fig. 29 and part of the deformed Fig. 30 were learned. Then, the learning result was applied to Figs. 28 and 29 and verified. Figures 33 and 34 show them solid angle values in Figs. 29 and 30 used for learning. In addition, the solid angle value in Figs. 31 and 32 are shown in Figs. 35 and 36. Details about Figs. 26 to 29 are given in Tables 3 and 4.

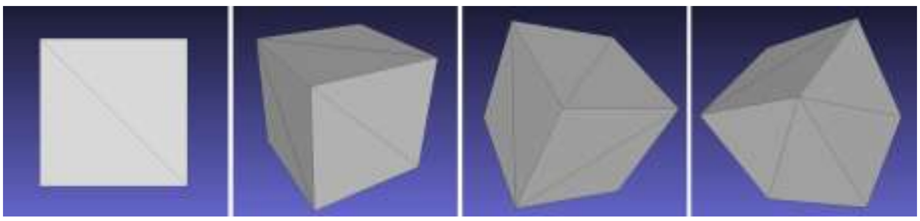

Fig. 9. Original shape (regular cube)

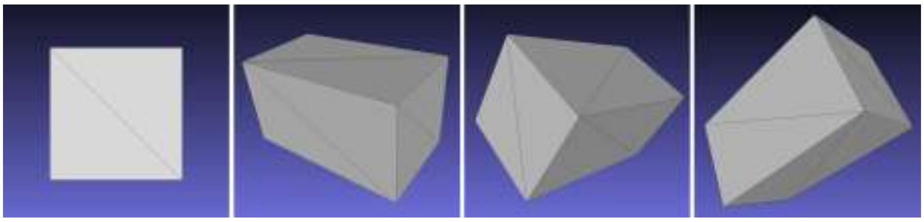

Fig. 10. Original shape (cuboid 1)

Fig. 11. Original shape (cuboid 2)

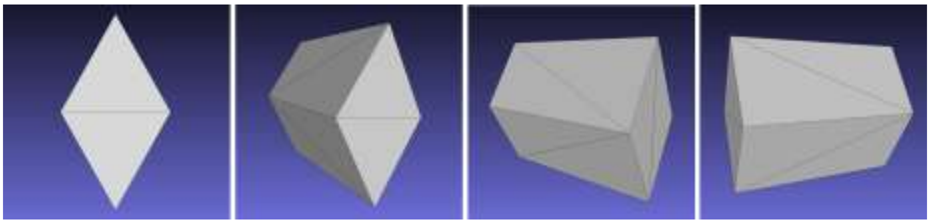

Fig. 12. Original shape (hexahedron)

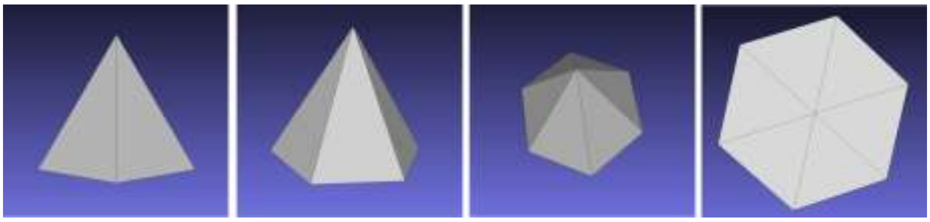

Fig. 13. Original shape (cone)

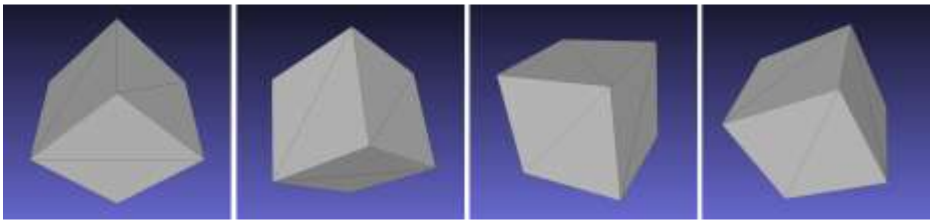

Fig. 14. Scale change plus rotation (regular cube)

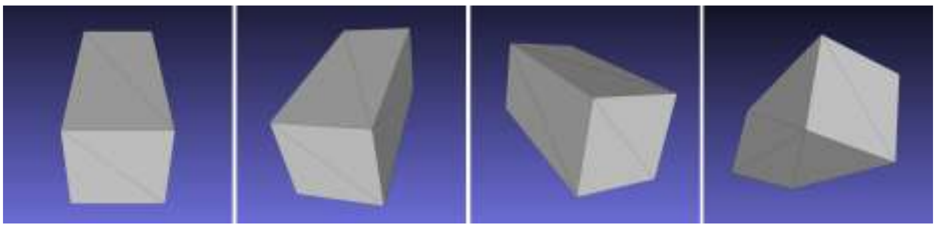

Fig. 15. Scale change plus rotation (cuboid 1)

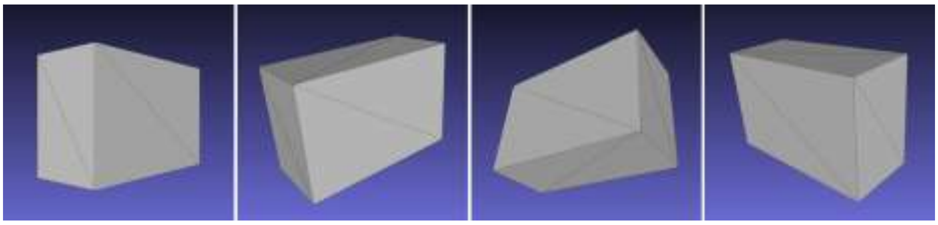

Fig. 16. Scale change plus rotation (cuboid 2)

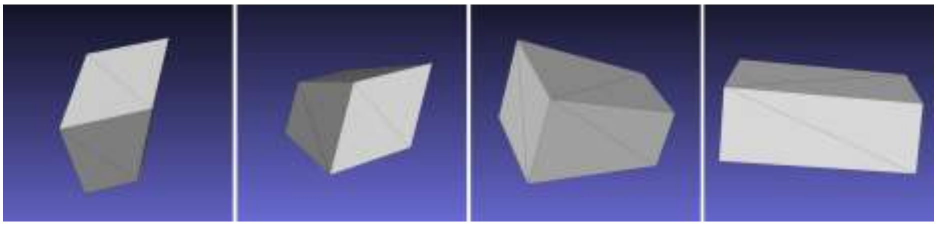

Fig. 17. Scale change plus rotation (hexahedron)

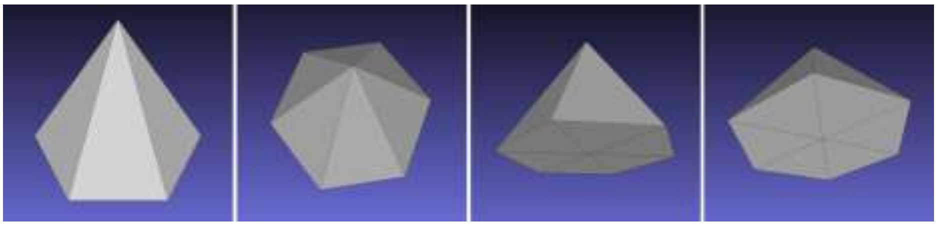

Fig. 18. Scale change plus rotation (corn)

Table. 2. Data set for experiments

	Category	H × W × L	Number of polygons
Fig. 9.	Cube	30 × 30 × 30	12
Fig. 10.	Cuboid 1	25 × 25 × 45	12
Fig. 11.	Cuboid 2	25 × 35 × 45	12
Fig. 12.	Hexahedron	20 × 35 × 45	12
Fig. 13.	Cone	25 × 25 × 40	12

Table. 3. Data set for experiments

	Category	H × W × L	Number of polygons	Remarks
Fig. 14.	Cube	20 × 20 × 20	12	Shrink + Rotation
Fig. 15.	Cuboid 1	20 × 20 × 45	12	Raito change (partial) + Rotation
Fig. 16.	Cuboid 2	30 × 40 × 55	12	Blow up + Rotation
Fig. 17.	Hexahedron	15 × 25 × 45	12	Ratio change (partial) + Rotation
Fig. 18.	Cone	20 × 20 × 30	12	Ratio change + Rotation

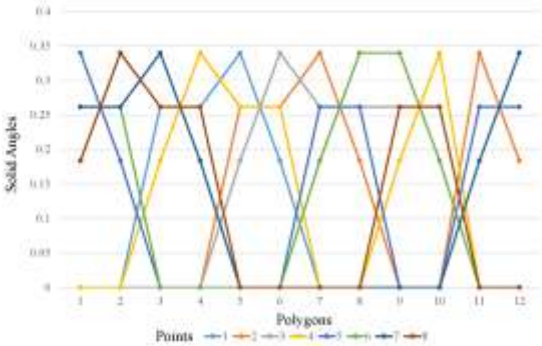

Fig. 19. Solid angle (regular cube (Fig. 9))

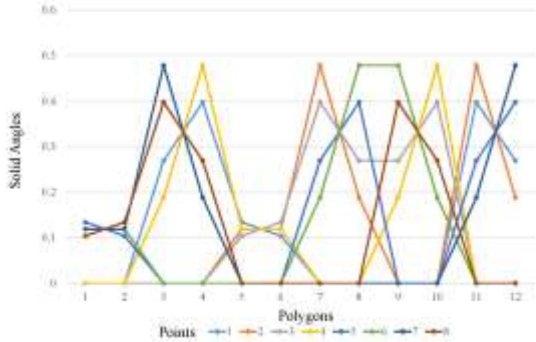

Fig. 20. Solid angle (cuboid 1 (Fig. 10))

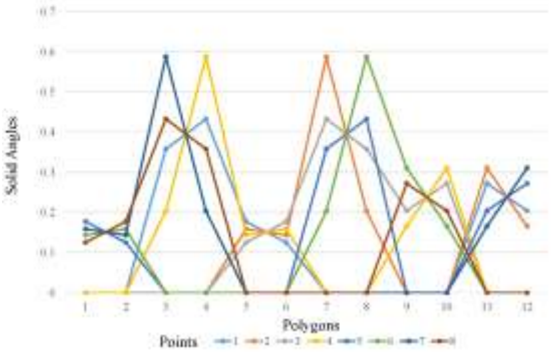

Fig. 21. Solid angle (cuboid 2 (Fig. 11))

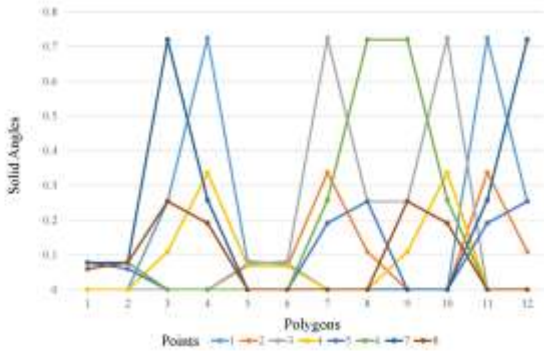

Fig. 22. Solid angle (hexahedron (Fig. 12))

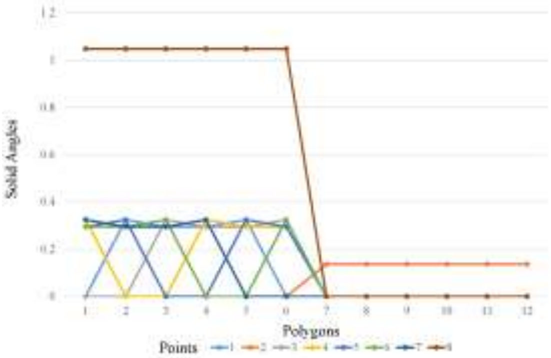

Fig. 23. Solid angle (cone (Fig. 13))

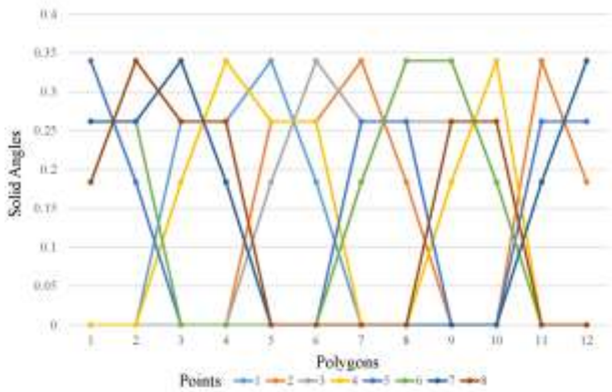

Fig. 24. Solid angle (regular cube (Fig. 14)

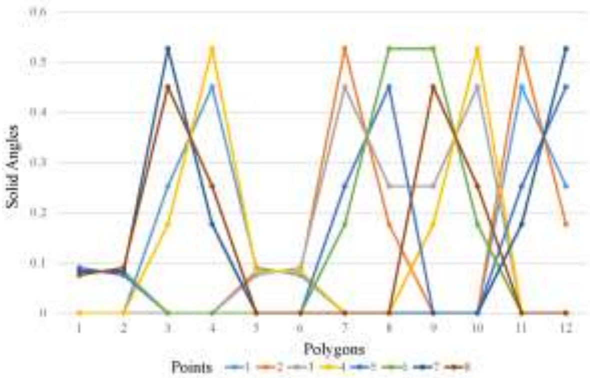

Fig. 25. Solid angle (cuboid 1 (Fig. 15))

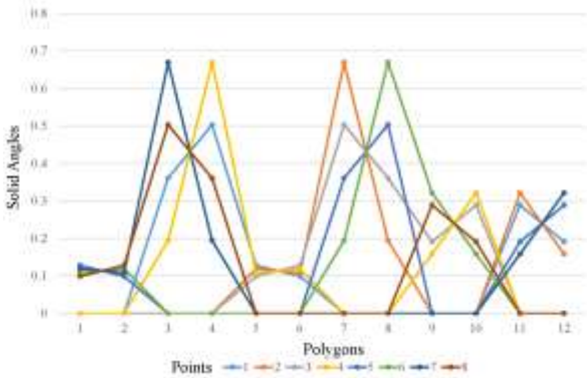

Fig. 26. Solid angle (cuboid 2 (Fig. 16))

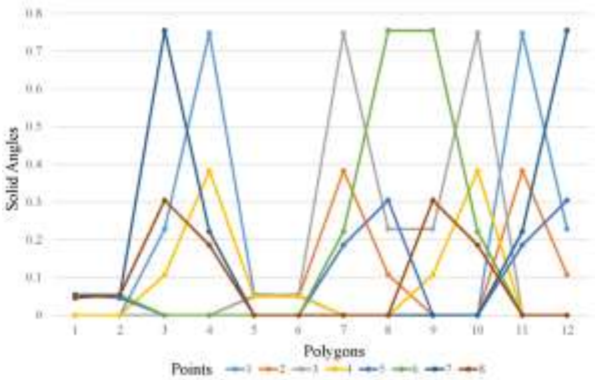

Fig. 27. Solid angle (hexahedron (Fig. 17))

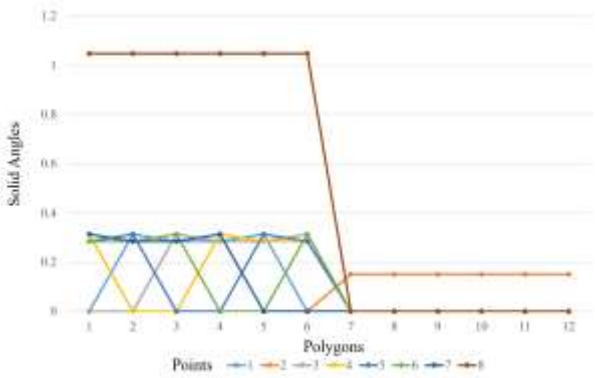

Fig. 28. Solid angle (cone (Fig. 18))

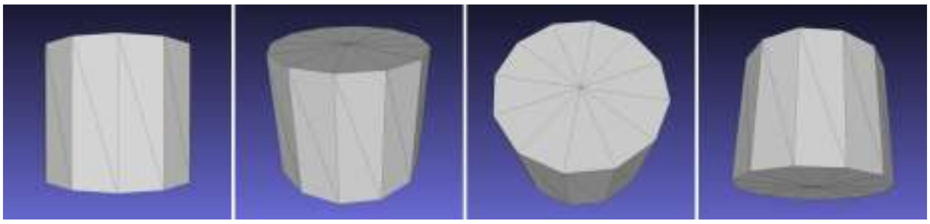

Fig. 29. Original shape (Cylinder)

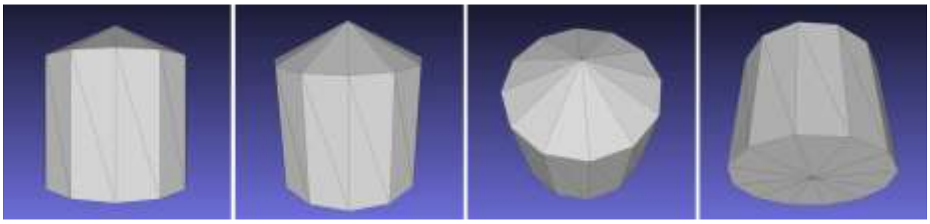

Fig. 30. Original shape (deformation)

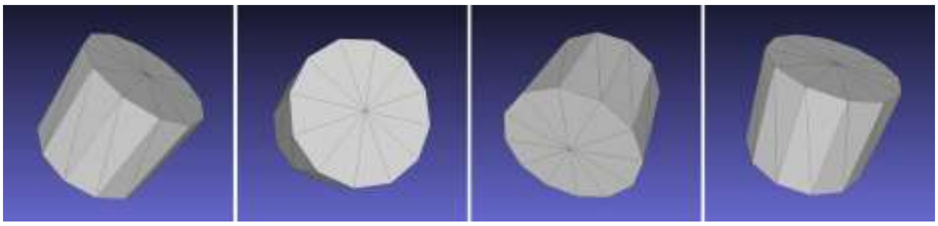

Fig. 31. Scale change plus rotation (Cylinder)

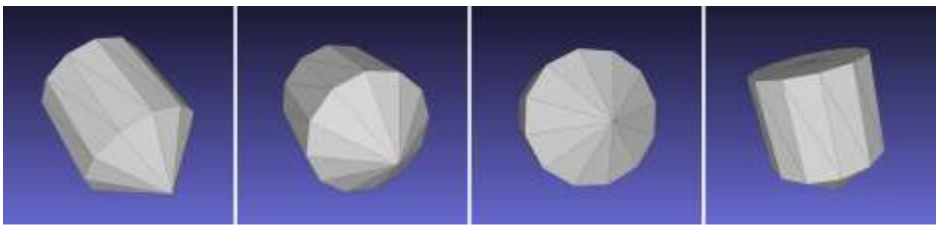

Fig. 32. Deformation and transformations (scale change plus rotation)

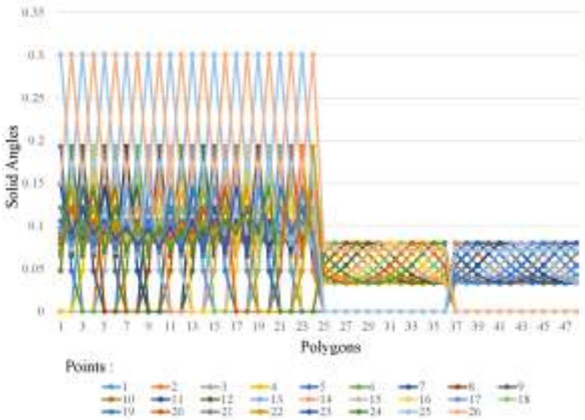

Fig. 33. Solid angle (Cylinder (Fig. 29))

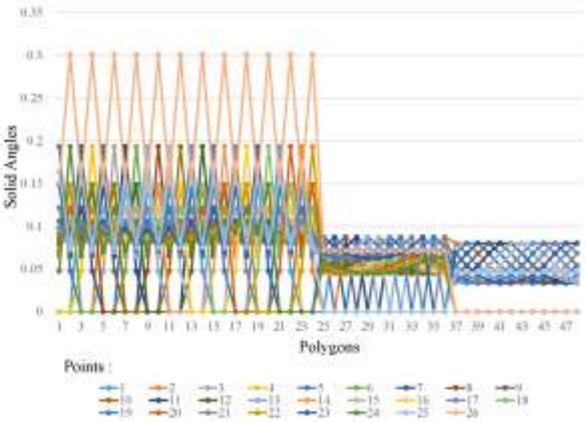

Fig. 34. Solid angle (Cylinder deformed (Fig. 30))

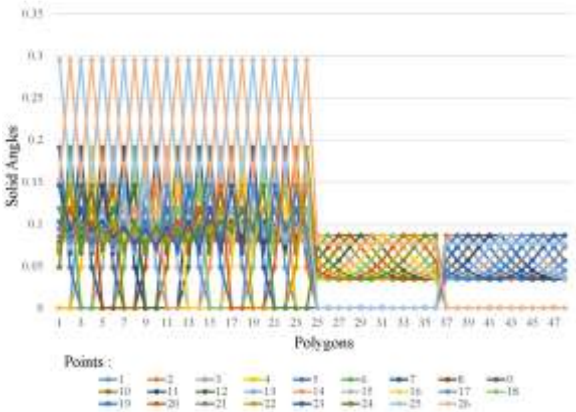

Fig. 35. Solid angle (Cylinder (Fig. 31))

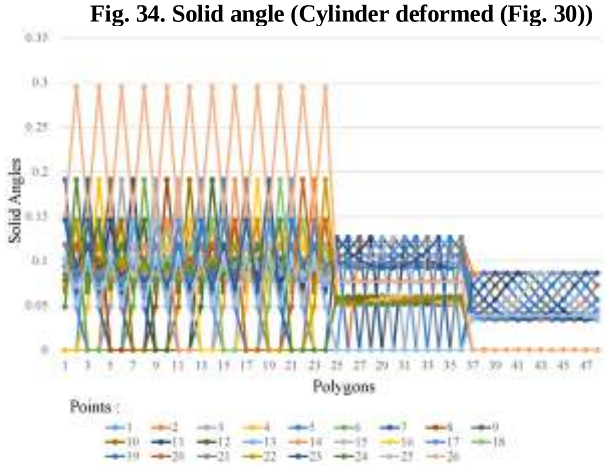

Fig. 36. Solid angle (Cylinder deformed (Fig. 32))

Table. 4. Shape to learn

	Category	H × W × L	Number of polygons
Fig. 29.	Cylinder	30 × 30 × 40	24
Fig. 30.	Cylinder(deformed)	30 × 30 × 35	24

Table. 5. Unlearned Shapes

	Category	H × W × L	Number of polygons	Remarks
Fig. 31.	Cylinder	25 × 25 × 35	24	Raito change + Rotation
Fig. 32.	Cylinder(deformed)	25 × 25 × 30	24	Raito change + Rotation

The environment used for learning and verification is summarized in Table 6.

Table. 6. Experimental environment

	Specification
CPU	Intel Core i7-5500U 2.40 GHz
RAM	8 GB DDR3L-RS 1600 MHz
Storage	SSD (SATA) 1 TB
OS	Windows 10 Professional
Development environment	javac 1.8.0_211

V. RESULTS

In our experiments, it was possible to perform determine on cuboids and cubes. In addition, the result that it was the same also about the similar cuboid was able to be shown. Further, it was possible to determine that the figure was the same shape relative to the figure obtained by rotating the cuboid.

The deformed figure of the cylinder and cylinder shows a different value in the shape determination. As a result, after learning each, relative to the unknown cylinder, it was possible to properly classify by applying the learning results.

From the above results, it is considered that a shape can be determined based on learned data for a new three-dimensional shape.

Note that the same modeling software was used in the experiment, the corresponding polygons were the same for both learning and verification. Based on a specific value (i.e., 0), by sorting , but can be partially supported, in the future, even when the polygon sequence to appear randomly, it is considered that it is necessary to perform verification of whether recognizable.

VI. CONCLUSION

It is considered possible to recognize a three-dimensional figure using a solid angle.

Generally, for three-dimensional shape learning, shape determination is performed using deep learning. However, in recent years, the black box nature of deep learning has become a concern, and there is demand for accountability in machine learning. In this study, it is a problem that the variable increases according to the number of polygons in the learning, but the feature is shown for each shape. Therefore, it is considered difficult to be a black box.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 19K14599.

REFERENCES

[1] Min Liu, Fupin Yao, Chiho Choi, Ayan Sinha, Karthik Ramani, "Deep Learning 3D Shapes Using Alt-az Anisotropic 2-Sphere Convolution," 7th International Conference on Learning Representations, 2019.

[2] AI Scholar, https://ai-scholar.tech/treatise/deeplearning-3d-36/, 2018.

[3] Daeyun Shin, Charless C. Fowlkes, Derek Hoiem, "Pixels, voxels, and views: A study of shape representations for single view 3D object shape prediction," Conference on Computer Vision and Pattern Recognition 2018, DOI: 10.1109/CVPR.2018.00323, 2018.

[4] David Griffiths, Jan Boehm, "A Review on Deep Learning Techniques for 3D Sensed Data Classification," Remote Sensing, 11(12), 1499, DOI: 10.3390/rs11121499, 2019.

[5] Katja Auernhammer, Ramin Tavakoli Kolagari, Markus Zoppelt, "Attacks on Machine Learning: Lurking Danger for Accountability," Conference on Artificial Intelligence 2019 (AAAI-19), 2019.

[6] Naveed Akhtar, Ajmal Mian, "Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey," in IEEE Access, vol. 6, pp. 14410-14430, DOI: 10.1109/ACCESS.2018.2807385, 2018.

[7] Tom Slee, "Algorithmic Accountability: The Other Side of Machine Learning," SAP SuccessFactors, https://www.successfactors.com/resources/knowledge-hub/algorithmic-accountability-the-other-side-of-machine-learning.html, 2019.

[8] Michael Fullan, Joanne Quinn, Joanne McEachen, "Deep Learning: Engage the World Change the World," Corwin, ISBN-10: 1506368581, 2017.

[9] Raju Gudhe, "Point Cloud Data for Deep Learning (Part_01)," https://medium.com/@graju1401/point-cloud-data-for-deep-learning-part-01-ecc1445b3728, Medium, 2018.

[10] Daniel Maturana, Sebastian Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 922-928, DOI: 10.1109/IROS.2015.7353481, 2015.

[11] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, Jianxiong Xiao, "3D ShapeNets: A Deep Representation for Volumetric Shapes," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), DOI: 10.1109/CVPR.2015.7298801, 2015.

[12] Nima Sedaghat, Mohammadreza Zolfaghari, Ehsan Amiri, Thomas Brox, "Orientation-boosted Voxel Nets for 3D Object Recognition," Proceedings of the British Machine Vision Conference (BMVC), DOI :10.5244/c.31.97, 2017.

[13] Jun Mitani, Hiromasa Suzuki, Hiroshi Uno: "Computer Aided Design for Origamic Architecture Models with Voxel Data Structure," IPSJ Journal Vol.44, No. 5, pp.1372-1379, 2003.

[14] Masanobu Nagase, Shuichi Akizuki, Manabu Hashimoto, "Extraction of 3-D Feature Point for effect in Object Recognition based on Local Shape Distinctiveness," Information Processing Society of Japan, IPSJ SIG Technical Report, Vol. 2013, No. 26, 2013.

[15] Hang Su, Subhransu Maji, Evangelos Kalogerakis, Erik Learned-Miller, "Multi-view Convolutional Neural Networks for 3D Shape Recognition," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 945-953, DOI: 10.1109/ICCV.2015.114, 2015.

[16] Haoxuan You, Yifan Feng, Rongrong Ji, Yue Gao, "PVNet: A Joint Convolutional Network of Point Cloud and Multi-View for 3D Shape Recognition," Proceedings of the 26th ACM international conference on Multimedia, pp. 1310-1318, DOI: 10.1145/3240508.3240702, 2018.

[17] Nakayama Atsushi, Kawakatsu Daisuke, Kobori Ken-ichi, Kutsuwa Toshirou, "A checking method for a point inside a polyhedron in grasping an object of VR," The 48th National Convention of Information Processing Society of Japan (IPSJ), 2, 297–298, 1994.

[18] Satoshi Kodama, "Effectiveness of inside/outside determination in relation to 3D non-convex shapes using CUDA," The Imaging Science Journal, Vol. 66(7), pp. 409-418, DOI: 10.1080/13682199.2018.1497251, 2018.

[19] Satoshi Kodama, Yuka Ozeki, Rei Nakagawa, "Internal and External Analysis Considering the Layers of Three-dimensional Shapes Using CUDA," International Journal of Computer Trends and Technology (IJCTT), Vol. 67(6), DOI: 10.14445/22312803/IJCTT-V67I6P101, 2019.

[20] C. Wayne Brown, "Volume / Surfaces," Learn WebGL, http://learnwebgl.brown37.net/model_data/model_volume.html, 2016.

[21] Justin A. Jensen, Robert P. Burton, "Fourveo: Integration of 4D animation into conventional 3D animation workflows," Computer Animation and Virtual Worlds, Vol. 29 (3-4), DOI: 10.1002/cav.1816, 2018.