Primitive Shape Recognition via Superquadric Representation using Large Margin Nearest Neighbor Classifier

Ryo Hachiuma, Yuko Ozasa and Hideo Saito
Graduate School of Science and Technology, Keio University, Yokohama, Japan
{*ryo-hachiuma, yuko.ozasa, hs*}*@keio.jp*

Keywords: Superquadrics, 3D Shape Primitives, Primitive Shape Recognition, Large Margin Nearest Neighbor.

Abstract: It is known that humans recognize objects using combinations and positional relations of primitive shapes. The first step of such recognition is to recognize 3D primitive shapes. In this paper, we propose a method for primitive shape recognition using superquadric parameters with a metric learning method, large margin nearest neighbor (LMNN). Superquadrics can represent various types of primitive shapes using a single equation with few parameters. These parameters are used as the feature vector of classification. The real objects of primitive shapes are used in our experiment, and the results show the effectiveness of using LMNN for recognition based on superquadrics. Compared to the previous methods, which used *k*-nearest neighbors (76.5%) and Support Vector Machines (73.5%), our LMNN method has the best performance (79.5%).

1 INTRODUCTION

Nowadays, due to the improvement of sensing technology, the 3D data of a scene or an object can easily be captured by a depth sensor, such as Kinect (Zhang, 2012). 3D object recognition is one of the important tasks in the field of computer vision for understanding scenes and robot manipulation.

It has been widely studied that how human recognize objects (Marr D, 1982; Biederman, 1987). It is said that a small number of fundamental primitives suffice to represent most objects for the purpose of generic recognition, and we recognize and decide the attributes of objects using the combinations and positional relations of these primitives (Biederman, 1987). Therefore, it is important to recognize objects by these primitives to achieve a recognition system based on human recognition.

To achieve primitives-based recognition, it is necessary to represent primitives with consistency and simplicity. One of the most appropriate representations is superquadrics (Barr, 1981). Superquadrics are one of the methods for primitive shape representation that can represent a variety of shapes with few parameters.

We focus on primitive shape-based object recognition using superquadric representation. There are two related works that recognize objects using superquadric representation: k-nearest neighbor (*k*NN) (Raja and Jain, 1992) and support vector machines (SVM) (Xing et al., 2004).

Recently, metric learning has been considered a high-performance classifying method. In particular, large margin nearest neighbor (LMNN) is a metric learning method with a training Mahalanobis distance metric for *k*NN classification (Weinberger and Saul, 2009). In this paper, we present a method that recognizes superquadric parameters with LMNN.

Before the low-cost depth sensor Kinect (Zhang, 2012) was developed, it was difficult to evaluate the results of object recognition using superquadric parameters with real objects. Raja, N. S. and Jain, A. K. (Raja and Jain, 1992) used real objects in their experiments. They captured one snapshot of each real object, and they recognize one superquadric parameter per object with Euclidean distance matching. Xing, W. *et.al.* (Xing et al., 2004) used only ideal parameters and did not use real objects in their experiments. They classify these ideal parameters with SVM. In our experiment, we captured hundreds of shots of real primitive objects with a depth sensor, and we constructed a primitive object dataset using Kinect(Zhang, 2012). This dataset enables us to evaluate in detail our method using real objects and compare our method with previous work (Raja and Jain, 1992; Xing et al., 2004). No previous studies have compared the classification results with *k*NN and SVM. Our experiments show that the proposed method with the LMNN classifier has the highest performance in comparison with SVM and *k*NN classi-

Hachiuma, R., Ozasa, Y. and Saito, H.
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) - Volume 5: VISAPP*, pages 325-332
ISBN: 978-989-758-226-4

fiers.

The novelty of this paper is to use LMNN for primitive shape recognition based on superquadrics. There are three contributions of this paper. First, a 3D dataset of real primitive objects is constructed. Second, the parameters of the superquadrics of real objects are estimated, and these parameters are used for the recognition. Third, we compare the results of three classifiers: *k*NN, SVM, and LMNN.

The rest of the paper is organized as follows. We review related work on 3D object recognition and superquadrics in the field of computer vision in Section 2. Section 3 describes our methods and our superquadric parameter estimation and recognition model. The experimental results and evaluations of recognition are shown in Section 4. Finally, Section 5 concludes the paper.

2 RELATED WORKS

We present primitive shape-based object recognition, and superquadrics are used for the primitive shape representation. In this section, we briefly introduce several related approaches to primitive shape-based object recognition. Moreover, as superquadrics have been used in many different ways in the field of computer vision, we also introduce these research.

2.1 3D Object Recognition based on Primitive Shapes

Due to the improvement of sensing technology, 3D object recognition methods have been studied. There are two major kinds of 3D object recognition methods. One is based on specific feature extraction and recognition, while the other is based on the human recognition system. The former method creates a histogram of normal vectors or relations of neighbors (Rusu et al., 2009; Tombari et al., 2010; Tang et al., 2012). These histogram-based approaches have high performance with specific object recognition or in cluttered scenes.

In the extensive literature on 3D object recognition, some studies discuss the human recognition system. For example, Nieuwenhuisen, M. *et.al.* proposed a robot grasping method by estimating cylindrical parameters of objects (Nieuwenhuisen et al., 2012). We employed superquadrics to represent primitive shapes, so that not only cylinders but also other primitives could be represented. Somani, N. *et.al.* present a method for specific object recognition that considers the physical constraints of the primitives' orientation (Somani et al., 2014). We do not consider the constraints of orientations. However, as we used a statistical machine-learning method for object recognition, our method has the potential to be implemented in genetic object recognition. It is necessary to segment objects into primitive shapes to achieve object recognition that is represented in several primitives. Garcia S. (Garcia, 2009) showed that the suitable segmentation algorithm for man made objects is segmentation based on primitive fitting or 3D volumetric approaches. In this paper, primitive-based object segmentation will be explored.

2.2 Superquadrics in the Field of Computer Vision

Superquadrics in the field of computer vision has been investigated since (Pentland, 1986), and the research of superquadrics is actively developed in around 1990s. studies on superquadrics in the field of computer vision were on superquadric parameter estimation based on depth images (Solina and Bajcsy, 1987). Solina and Bajcsy presented a method for recognizing pieces of mail using superquadric parameters estimated from range images.

Thus, superquadrics have been used for object shape approximation (Strand et al., 2010; Solina and Bajcsy, 1990; Saito and Kimura, 1996), novelty detection (Drews Jr. et al., 2010), object segmentation (Chevalier et al., 2003; Leonardis et al., 1997), object grasping (Varadarajan and Vincze, 2011), and collision detection (Moustakas et al., 2007).

However, only few studies have conducted primitive shape recognition using superquadric representation (Xing et al., 2004; Raja and Jain, 1992). Raja, N. S. and Jain, A. K. (Raja and Jain, 1992) experimented with crafted primitive objects using *k*NN, while Xing, W. *et.al.* (Xing et al., 2004) experimented with ideal superquadric parameters using SVM. In this paper, we use superquadric parameters for object recognition, and we employ the classifier LMNN. We experimented with real primitive shapes.

3 PRIMITIVE SHAPE-BASED RECOGNITION

In this section, we introduce a method to recognize primitives based on superquadric representation. Our method in this paper consists of 2 main steps: First, superquadric parameter is estimated from 3D data points of the object. Second, we set these estimated parameters as a feature vector $\boldsymbol{F}$ and recognize $\boldsymbol{F}$ using LMNN.

3.1 Superquadrics

Superquadrics are an extension of quadric surfaces those include superellipsoids, supertoroids, and superhyperboloids. Superquadrics have been proposed for use as primitives for shape representation in the field of computer graphics (Barr, 1981) and computer vision (Pentland, 1986). A superquadric surface can be defined by the 3D vector **x**.

$$\mathbf{x}(\eta,\omega) = \begin{bmatrix} a_1 cos^{\varepsilon_1}(\eta) cos^{\varepsilon_2}(\omega) \\ a_2 cos^{\varepsilon_1}(\eta) sin^{\varepsilon_2}(\omega) \\ a_3 sin^{\varepsilon_1}(\eta) \end{bmatrix}, \quad (1)$$

$$-\pi/2 \leq \eta \leq \pi/2, -\pi \leq \omega \leq \pi.$$

The surface of superquadrics is located in the original coordinate system. In Eq. (1), there are two independent variables: η and ω. Parameter η is the angle that expresses between the x-axis and the projection of vector **x** in the x-y plane, and parameter ω is the angle that expresses between vector **x** and its projection to the x-y plane. Eq. (1) can be written in implicit form as Eq. (2) by eliminating two parameters: η and ω.

$$f(x_s,y_s,z_s) = \left\{ \left(\frac{x_s}{a_1}\right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y_s}{a_2}\right)^{\frac{2}{\varepsilon_2}} \right\}^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z_s}{a_3}\right)^{\frac{2}{\varepsilon_1}} = 1. \quad (2)$$

In Eq. (2), there are five parameters $(a_1,a_2,a_3,\varepsilon_1,\varepsilon_2)$. Parameters a_1,a_2,a_3 are scale parameters that define the superquadric size in the x, y, and z coordinates, respectively, and parameters ε_1 and ε_2 are shape representation parameters that express squares along the z axis and the x-y plane, while the index s denotes that the point belongs to the superquadric-centered coordinate. Fig. 1 shows various superquadrics by changing the shape-representation parameters ε_1 and ε_2.

As shown in Fig. 1, the superquadric surface is shaped like a cylinder when $\varepsilon_1 \ll 1$ and $\varepsilon_2 = 1$, is shaped like a cuboid when $\varepsilon_1 \ll 1$ and $\varepsilon_2 \ll 1$, and is shaped like a sphere when $\varepsilon_1 = 1$ and $\varepsilon_2 = 1$.

3.1.1 Coordinate Transformation

As explained, the superquadric surface is located in its original coordinate system, but the input 3D point-cloud from the depth sensor is located in the world coordinate system. We want to transform a point from the world coordinate system to the superquadric-centered coordinate system with transformation matrix **T**. A relationship between the world coordinate

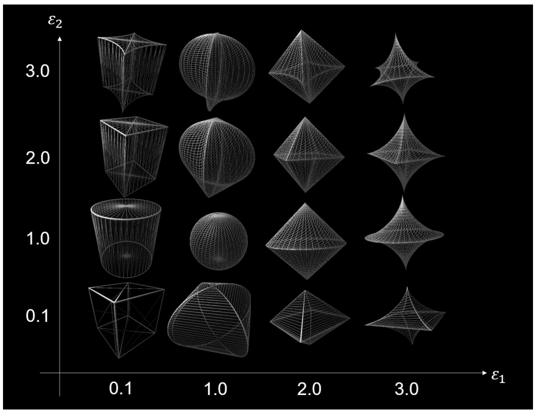


Figure 1: The various superquadric shapes according to ε_1 and ε_2.

system (x_w,y_w,z_w) and the superquadric-centered coordinate system (x_s,y_s,z_s) can be expressed in the following Eq. (3), where the transformation matrix **T** can be decomposed in rotation $\mathbf{R}(\theta_x,\theta_y,\theta_z)$ and translation $\mathbf{t}(t_x,t_y,t_z)$.

$$\begin{pmatrix} x_s \\ y_s \\ z_s \\ 1 \end{pmatrix} = (\mathbf{R}|\mathbf{t}) \begin{pmatrix} x_w \\ y_w \\ z_w \\ 1 \end{pmatrix}. \quad (3)$$

3.1.2 Superquadric Parameter Estimation

Superquadrics can be expressed in implicit form, which is called "inside-outside" function. When $f(x_s,y_s,z_s) > 1$, the point $p(x_s,y_s,z_s)$ lies outside of the surface, and if $f(x_s,y_s,z_s) = 1$, the point lies on the surface, and if $f(x_s,y_s,z_s) < 1$, the point lies inside the surface. If a point (x_w,y_w,z_w) from the world coordinate is given, Eq. (2) can be re-written with five parameters for superquadrics and six for transformation.

Given a set of N unstructured 3D data points, we want to estimate 11 parameters that the input 3D data points will fit or be close to the superquadrics model surface. As the superquadric surface must satisfy Eq. (2), we want to minimize Eq. (4) to estimate superquadric parameters.

$$\sum_{i=0}^{N} (f(x_{w_i},y_{w_i},z_{w_i}) - 1)^2. \quad (4)$$

However, 3D data points captured from a depth sensor have self-occlusion, and tremendous size differences of superquadrics can be fit into Eq. (4). It is known that by multiplying the square root of the scale parameters $\sqrt{a_1a_2a_3}$ to minimize the function described in Eq. (4), it is possible to estimate the smallest superquadrics by fitting the following 3D data points.

$$\sum_{i=0}^{n}(\sqrt{a_1a_2a_3}(f(x_{w_i},y_{w_i},z_{w_i})-1))^2. \quad (5)$$

Since function $\sqrt{a_1a_2a_3}(f(x_{w_i},y_{w_i},z_{w_i})-1)$ is a nonlinear function of 11 parameters, it is well known to solve function in Eq. (5) as a nonlinear least squares problem, in particular by using the Levenberg-Marquardt algorithm (Press et al., 1986). Moreover, when $\varepsilon_1,\varepsilon_2 < 0.1$, the inside-outside function will be numerically unstable, and as shown in Fig. 1, the superquadric will have concavities when $\varepsilon_1,\varepsilon_2 > 2.0$. We use the constraints when minimizing the function in Eq. (5) for the shape parameters: $0.1 < \varepsilon_1,\varepsilon_2 < 2.0$ and for the scale parameters : $a_1,a_2,a_3 > 0.0$. However, it is important to decide the initial parameters because this decision will involve the local minimum. We will explain the setting of the initial parameters in the next section.

3.1.3 Finding the Initial Parameters

As the function in Eq. (5) is not a convex function,the initial parameters will determine which local minimum the minimization will converge. It is important to estimate the rough parameters: translation, rotation, scale, and shape parameters.

First, as it is difficult to roughly estimate the shape of the object, the initial shape parameters ε_1 and ε_2 are set to 1, which means that the shape of the initial model is an ellipsoid. Second, the centroid of all 3D data points can be used to estimate the initial translation. Third, to compute the initial rotation, we compute the covariance matrix of all n 3D data points. From this covariance matrix, three pairs of eigenvectors and eigenvalues can be computed. The largest eigenvector of the covariance matrix always points in the direction of the largest variance of the data, and the magnitude of this largest vector equals the corresponding eigenvalue. The second largest eigenvector is always orthogonal to the largest and points in the direction of the second largest spread of the data which is the same as the third vector. Therefore, the eigenvectors can be used as the initial rotation parameters, and the eigenvalues can be used as the initial scale parameters.

3.2 Recognition System

In this paper, we set five superquadric parameters $(\varepsilon_1,\varepsilon_2,a_1,a_2,a_3)$ as a feature vector $\boldsymbol{F} = (\varepsilon_1,\varepsilon_2,a_1,a_2,a_3)$ that is estimated from the 3D data points of the object, and we apply the statistical machine-learning method to recognize objects.

Let $\{(\boldsymbol{F}_i,l_i)\}_{i=1}^{L\times N}$ denote a training set of $L\times N$ labeled examples with inputs $\boldsymbol{F}\in\mathbb{R}^5$ and discrete class labels y_i. L stands for the number of classes, and N stands for the number of examples per class through the rest of the paper. LMNN (Weinberger and Saul, 2009) is one of the metric learning methods that trains a Mahalanobis distance metric for kNN classification. Let covariance matrix M and the Mahalanobis distance $D_M(\boldsymbol{F}_p,\boldsymbol{F}_q)$ between two inputs, $\boldsymbol{F}_p$ and $\boldsymbol{F}_q$, be defined as follows.

$$D_M(\boldsymbol{F}_p,\boldsymbol{F}_q)=\sqrt{(\boldsymbol{F}_p-\boldsymbol{F}_q)^T M(\boldsymbol{F}_p-\boldsymbol{F}_q)} \quad (6)$$

where p and q denote the target indices, and $0 \le \{p,q\} \le L\times N$.

This metric is optimized with the goal that kNN always belongs to the same class, while data from different classes are separated by a large margin. It means that minimizing the Mahalanobis distance between a target data $\boldsymbol{F}_i$ and data that belongs to the same class of $\boldsymbol{F}_i$, maximizing the Mahalanobis distance between $\boldsymbol{F}_i$ and data belongs to different classes of $\boldsymbol{F}_i$. Hence, input data will be able to classify with accuracy. However, the computation cost will be extraordinarily huge if it computes the Mahalanobis distance between a target data $\boldsymbol{F}_i$ and all data belonging to the same class. LMNN uses k target neighbors to reduce the computation cost. Target neighbor is the k nearest data to the $\boldsymbol{F}_i$ with the same class. Moreover, it uses the idea of a large margin. A margin is a unit that separates data with different classes. As above, matrix M is learned as an optimization problem with Eq. (7).

$$\begin{aligned}&\textbf{Minimize} \sum_{ij}\eta_{ij}D_M(\boldsymbol{F}_i,\boldsymbol{F}_j)+c\sum_{ijh}(1-\delta_{ih})\xi_{ijh}\\&\textbf{subject to:}\\&D_M(\boldsymbol{F}_i,\boldsymbol{F}_h)-D_M(\boldsymbol{F}_i,\boldsymbol{F}_j)\ge 1-\xi_{ijh}\\&\xi_{ijh}\ge 0\\&M\succeq 0.\end{aligned} \quad (7)$$

where ηij denotes an indicate function $\eta\in\{0,1\}$ whether input $\boldsymbol{F}_j$ is a target neighbor of input $\boldsymbol{F}_i$ or not, and δ_{ih} also denotes an indicate function whether label l_i is the same class to l_h or not. ξ_{ijh} are the slack variables. c is a constant value range of 0 to 1.

The first term in the minimize function is the sum of the Mahalanobis distance between input $\boldsymbol{F}_i$ and k target neighbors, and the second term is the penalty term because it returns a positive value when the distance with the same label is smaller than with the different label.

4 EXPERIMENTS

We conducted two main experiments in order to evaluate and analyze the superquadric parameter estimation and effectiveness of LMNN. In Section 4.1, the superquadric parameters of five kinds of daily objects are estimated, and the appropriateness of the estimated parameters are evaluated. In Section 4.2, recognition using estimated parameters is evaluated. The proposed LMNN method is compared with the methods using *k*NN and SVM.

4.1 Superquadrics Estimation

First, we evaluated the results of the superquadric parameter estimation. We captured five objects using Kinect v.1 (Zhang, 2012) as a depth camera in this paper. Each object was put on the floor, and we created the 3D points of the object by extracting the 3D data points of the floor using the RANSAC algorithm (Schnabel et al., 2007), and the isolated points were extracted using Euclidean clustering (Rusu and Cousins, 2011). Fig. 2 shows the results of the estimation. In this figure, $(1a),(2a),\cdots,(5a)$ shows the RGB image of the objects captured from Kinect v.1, and $(1b),(2b),\cdots,(5b)$ shows the estimated superquadric surface.

Each estimated superquadric parameter is in the list in Fig. 2. We can compare these estimated parameters and look on Fig. 1 to see which superquadric shape that is correspond to them. For example, shape parameters of object $(1a)$ is $(\varepsilon_1,\varepsilon_2) = (0.10, 0.18)$, and it represents a cube in Fig. 1. Moreover, scale parameters of object $(1a)$ $(a_1,a_2,a_3) = (0.12, 0.07, 0.03)$ are reasonable because the size of this object is width = 22.8 cm, length = 11.7 cm, and height = 5.1 cm, and the ratio of each side is width:length:height = 4.47:2.29:1.0.

4.2 Recognition with Real Objects

The experiments conducted to check the performance of the SVM, *k*NN, and LMNN classifiers for superquadric classification in this section. Five types of primitive-shaped objects were used for the experiments $(L = 5)$. The objects were cuboid (width = 20 cm, length = 20 cm, and height = 10 cm), large cube (width = length = height = 25 cm), pyramid (width = length = 20 cm), cylinder (radius = 10 cm, height = 20 cm), and small cube (width = length = height = 20 cm). These objects are shown in Fig. 3.

We constructed a dataset by capturing 240 data ($N = 240$) per object placed on the floor. Each of these data were taken from different angle and positions. The 240 data were split into 160 data for training and 80 for test. To evaluate the effectiveness of the superquadrics parameter, three different feature sets ($\boldsymbol{F}_1, \boldsymbol{F}_2$, and $\boldsymbol{F}_3$) were selected for our comparison experiments.

Table 1: Classification accuracy (%) with SVM.

	SVM		
	Linear	Poly	RBF
F_1	45.3	51.2	**62.3**
F_2	42.5	38.8	**52.0**
F_3	60.8	64.2	**73.5**

Table 2: Classification accuracy (%) with *k*NN and LMNN (k = 3,5).

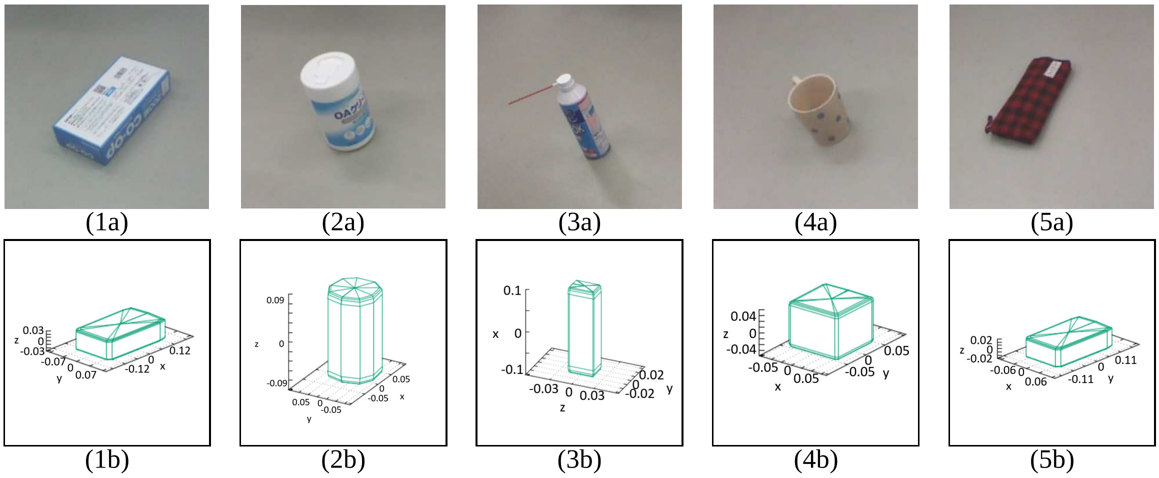
	*k*NN		LMNN	
	k=3	k=5	k=3	k=5
F_1	**62.2**	60.5	60.0	60.2
F_2	**68.1**	68.1	65.4	65.4
F_3	74.7	76.5	78.9	**79.5**

$$\begin{cases} \boldsymbol{F}_1 = (\varepsilon_1, \varepsilon_2), \\ \boldsymbol{F}_2 = (a_1, a_2, a_3), \\ \boldsymbol{F}_3 = (\varepsilon_1, \varepsilon_2, a_1, a_2, a_3). \end{cases} \quad (8)$$

Linear SVM and non-linear SVM (SVM with polynomial kernel and RBF kernel) are evaluated in this paper. Tab(1) shows that the SVM with an RBF kernel has the best performance for feature vector $\boldsymbol{F}_1, \boldsymbol{F}_2$, and $\boldsymbol{F}_3$, and it shows that feature vector $\boldsymbol{F}_3$ has the best performance with kernels, which means that not only the shape parameters but also the scale parameters have to be effective for superquadric object recognition. Comparing $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$, $\boldsymbol{F}_1$ has better performance than $\boldsymbol{F}_2$. This means that shape parameters are more valid than scale parameters. Fig. 4 (a) shows the confusion matrix of the SVM classifier when the set of $C = 100.0, \gamma = 0.1$. C and γ are parameters of SVM with the RBF kernel, and these parameters are optimized in our experiment.

Second, the experimental results for the *k*NN and the LMNN classifier are shown in Tab. 2 with neighborhood size $k = 3$. Fig. 4 (b) and (c) show the confusion matrices of the *k*NN and LMNN classifiers when $k = 3$. LMNN has the best performance (79.5%) in comparison with the *k*NN (76.5%) and SVM (73.5%) classifier with feature vector $\boldsymbol{F}_3$.

Fig. 5 shows the scatter plot of estimated parameters ε_1 and ε_2 for each object shown in Fig. 3. We will take a closer look at Fig. 5 and Fig. 1. Fig. 5(4) shows a scatter plot of ε_1 and ε_2 for the



$$(1)(\varepsilon_1,\varepsilon_2) = (0.10, 0.18), (a_1,a_2,a_3) = (0.12, 0.07, 0.03)$$
$$(2)(\varepsilon_1,\varepsilon_2) = (0.10, 0.80), (a_1,a_2,a_3) = (0.05, 0.05, 0.09)$$
$$(3)(\varepsilon_1,\varepsilon_2) = (0.10, 0.10), (a_1,a_2,a_3) = (0.10, 0.02, 0.03)$$
$$(4)(\varepsilon_1,\varepsilon_2) = (0.12, 0.10), (a_1,a_2,a_3) = (0.05, 0.05, 0.04)$$
$$(5)(\varepsilon_1,\varepsilon_2) = (0.10, 0.10), (a_1,a_2,a_3) = (0.06, 0.11, 0.02)$$

Figure 2: Five objects and the results of the superquadric parameter estimation (a: RGB Image object 3D data points b: Superquadric surface).

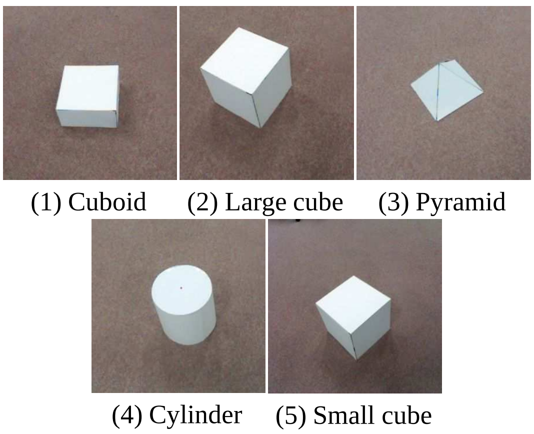


Figure 3: Primitive shapes used in the recognition section.

cylinder. Plots are concentrated in the vicinity of $(\varepsilon_1 = 0.1, \varepsilon_2 = 1)$, and $(\varepsilon_1 = 0.1, \varepsilon_2 = 1)$ which also represents a cylinder in Fig. 1. As the small cube and large cube are the same shape on a different scale, they have similar scatter plots. More interestingly, most of the parameters are distributed in the vicinity of $(\varepsilon_1 = 2.0, 0.1 < \varepsilon_2 < 2.0)$, the parameter represents an octahedron. For object (3), the pyramid, the superquadrics cannot represent it(tetrahedron), so the parameter of the pyramid is scattered in the vicinity of the octahedron. As we explained in Section 4.1, there is a double representation of the cube, and the parameters of the object (1), the cuboid, are distributed in $(\varepsilon_1 = 0.1, \varepsilon_2 = 0.1)$ and $(\varepsilon_1 = 0.1, \varepsilon_2 = 2.0)$. This double representation will be a crucial issue for recognition accuracy.

5 CONCLUSIONS AND FUTURE WORK

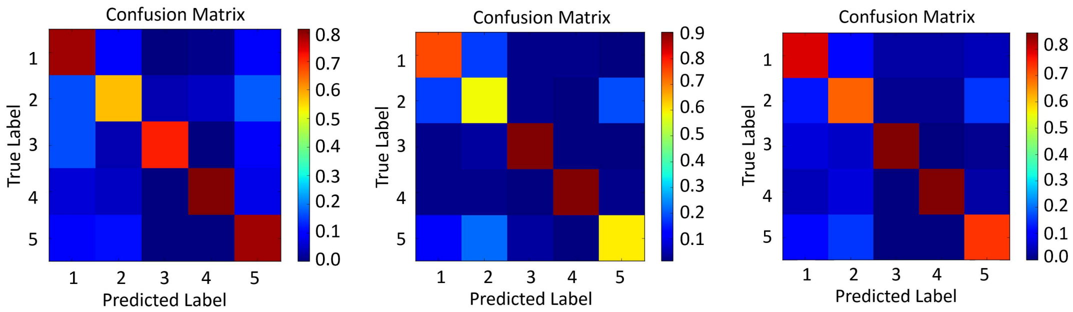
We proposed a method for recognizing primitive shapes that are represented in superquadric parameters using LMNN. The main novelty of this paper is as follows:

- Applying the metric learning method LMNN for primitive shape recognition with superquadric parameters.

The main contributions of this paper are as follows:

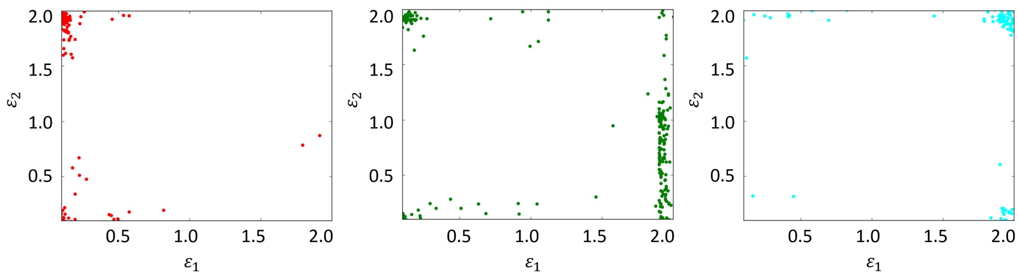
- Creating a the 3D primitive dataset captured by an depth sensor.
- Estimating superquadric parameters for real daily objects.
- Comparing the classification result for *k*NN, SVM, and LMNN.

We evaluated the result of superquadric parameter estimation, and recognition with these parameters. We evaluated the two main experiments in order to analyze superquadric parameter estimation and effectiveness of LMNN. The superquadric parameters of five kinds of daily objects were estimated, and the appropriateness of the estimated parameters were eval-

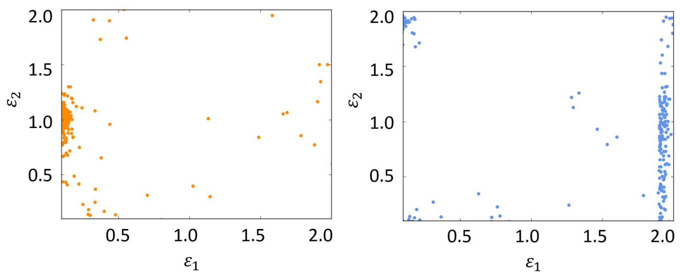


(a) Confusion matrix with SVM. (b) Confusion matrix with kNN. (c) Confusion matrix with LMNN.

Figure 4: Comparison of the confusion matrices of each classifier.



(1) Cuboid (2) Large cube (3) Pyramid



(4) Cylinder (5) Small cube

Figure 5: Scatter Plot of $(\varepsilon_1, \varepsilon_2)$ for each object shown in Fig. 3.

uated. The recognition using the estimated parameters was evaluated. The proposed LMNN method was compared with kNN and SVM, and the results showed that LMNN had the best performance (79.5%).

In future work, based on the human recognition system, we plan to segment objects into primitive shapes so that objects can be represented in combinations. Although there is a method for superquadric-based segmentation (Leonardis et al., 1997), it is difficult to capture the dense 3D data of objects in real environment. Therefore, we have to present a new method that can segment the sparse 3D data of objects into primitive shapes.

ACKNOWLEDGEMENTS

This work was partially supported by MEXT/JSPS Grant-in-Aid for Scientific Research(S) 24220004, and JST CREST Intelligent Information Processing Systems Creating Co-Experience Knowledge and Wisdom with Human-Machine Harmonious Collaboration

REFERENCES

Barr, A. H. (1981). Superquadrics and angle-preserving transformations. *IEEE Computer graphics and Applications*, 1(1):11–23.

Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167.

Chevalier, L., Jaillet, F., and Baskurt, A. (2003). Segmentation and superquadric modeling of 3d objects. In *WSCG*.

Drews Jr., P., Trujillo, P. N., Rocha, R. P., Campos, M. F. M., and Dias, J. (2010). Novelty detection and 3d shape retrieval using superquadrics and multi-scale sampling for autonomous mobile robots. In *ICRA*, pages 3635–3640.

Garcia, S. (2009). Fitting primitive shapes to point clouds for robotic grasping. *Master of Science Thesis. School of Computer Science and Communication, Royal Institute of Technology, Stockholm, Sweden*.

Leonardis, A., Jaklic, A., and Solina, F. (1997). Superquadrics for segmenting and modeling range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1289–1295.

Marr D, V. (1982). A computational investigation into the human representation and processing of visual information.

Moustakas, K., Tzovaras, D., and Strintzis, M. G. (2007). Sq-map: Efficient layered collision detection and haptic rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):80–93.

Nieuwenhuisen, M., Stückler, J., Berner, A., Klein, R., and Behnke, S. (2012). Shape-primitive based object recognition and grasping. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–5. VDE.

Pentland, A. P. (1986). Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(3):293–331.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1986). Numerical recipes: the art of scientific computing. *Cambridge U. Press, Cambridge, MA*.

Raja, N. S. and Jain, A. K. (1992). Recognizing geons from superquadrics fitted to range data. *Image and vision computing*, 10(3):179–190.

Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE.

Rusu, R. B. and Cousins, S. (2011). 3d is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4. IEEE.

Saito, H. and Kimura, M. (1996). Superquadric modeling of multiple objects from shading images using genetic algorithms. In *Industrial Electronics, Control, and Instrumentation, 1996., Proceedings of the 1996 IEEE IECON 22nd International Conference on*, volume 3, pages 1589–1593. IEEE.

Schnabel, R., Wahl, R., and Klein, R. (2007). Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library.

Solina, F. and Bajcsy, R. (1987). Range image interpretation of mail pieces with superquadrics.

Solina, F. and Bajcsy, R. (1990). Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE transactions on pattern analysis and machine intelligence*, 12(2):131–147.

Somani, N., Cai, C., Perzylo, A., Rickert, M., and Knoll, A. (2014). Object recognition using constraints from primitive shape matching. In *International Symposium on Visual Computing*, pages 783–792. Springer.

Strand, M., Xue, Z., Zoellner, M., and Dillmann, R. (2010). Using superquadrics for the approximation of objects and its application to grasping. In *Information and Automation (ICIA), 2010 IEEE International Conference on*, pages 48–53. IEEE.

Suykens, J. A. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300.

Tang, S., Wang, X., Lv, X., Han, T. X., Keller, J., He, Z., Skubic, M., and Lao, S. (2012). Histogram of oriented normal vectors for object recognition with a depth sensor. In *Asian conference on computer vision*, pages 525–538. Springer.

Tombari, F., Salti, S., and Di Stefano, L. (2010). Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer.

Varadarajan, K. M. and Vincze, M. (2011). Object part segmentation and classification in range images for grasping. In *Advanced Robotics (ICAR), 2011 15th International Conference on*, pages 21–27. IEEE.

Weinberger, K. Q. and Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244.

Xing, W., Liu, W., and Yuan, B. (2004). Superquadric-based geons recognition utilizing support vector machines. In *Signal Processing, 2004. Proceedings. ICSP'04. 2004 7th International Conference on*, volume 2, pages 1264–1267. IEEE.

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10.