

IMPROVING LOCAL SYMMETRY ESTIMATIONS IN RGB-D IMAGES BY FITTING SUPERQUADRICS

David Fornas Pedro J. Sanz
Interactive & Robotic Systems, Universitat Jaume I
Avda. Vicente Sos Baynat s/n, 12071 Castellón, Spain

Josep M. Porta Federico Thomas
Institut de Robòtica i Informàtica Industrial, UPC-CSIC
Llorens Artigas 4-6, 08028 Barcelona, Spain

Abstract

Real-time manipulation tasks rely on finding good candidates for apprehension points which, in turn, usually requires the computation of local symmetries. When RGB-D images are used as input information, these local symmetries can be deduced from segmenting these images and computing geometric moments for each cluster of points. This approach gives a rough approximation because it does not take into account that the considered points lie on a surface. In this paper, to improve the quality of the symmetry estimations, we propose a simple refinement process that takes as input the estimation obtained using moments and then fits a superquadric to the considered set of points. We evaluate our approach on data collected using a Microsoft's Kinect 2 sensor. The obtained experimental results demonstrate the efficacy of the proposed approach.

Keywords: Symmetry detection, Point clouds, Object Segmentation, Superquadrics, Kinect sensor.

1 INTRODUCTION

The Microsoft Kinect 2, introduced in 2014, is a cheap and powerful sensor able to deliver dense and reliable RGB-D images for almost any scene [19]. Such images open a wide range of applications in Robotics and, in particular in manipulation tasks.

In general, robust 3D geometric shape recovery and pose estimation is required to perform robotic manipulation (see, for example, [15, 17]). Nevertheless, due to its computational burden, real-time tasks rely on simply finding good candidates for apprehension points [3]. Determining such

points requires the computation of a set of geometric characteristics from the RGB-D image points that are likely to belong to the same object. The characteristics to compute depend on the kind of hand to be used, but one common set of features are symmetries, whose computation is classically addressed using moments [18]. At this point, it is important to distinguish between the moments of a surface patch and those corresponding to a set of points resulting from sampling this surface. The latter tend to the former as the number of points is increased, provided that they are evenly distributed on the surface. Otherwise, an error is present. Next, assuming that the robotic task is limited to the manipulation of man-made objects that can be approximated by spheres, cylinders and cuboids, we propose a refinement method to compute the moments of the object's surfaces. The method takes as initial estimate the moments obtained from the set of depth points and approximates the segmented regions by superquadrics.

The depth map delivered by the Kinect 2 sensor allows, after segmentation based on surface normals, to robustly fit surface patches to parts of the depth map. Simple shape primitives like planes, cylinders, or spheres are fast to compute and they capture a good range of typical man-made objects. The Point Cloud Library (PCL) [16] contains modules able to deal with these simple shapes. Nevertheless, superquadric shapes offer a greater flexibility in approximating a wide range of objects including the mentioned shape primitives using few parameters (see Figure 1).

This paper is organized as follows. First, Section 2 describes the object segmentation and the approximation of each object by a superquadric. Then, Section 3 presents the results of applying this method to a representative test case. Finally, Section 4 summarizes the work and enumerates points deserving further attention.

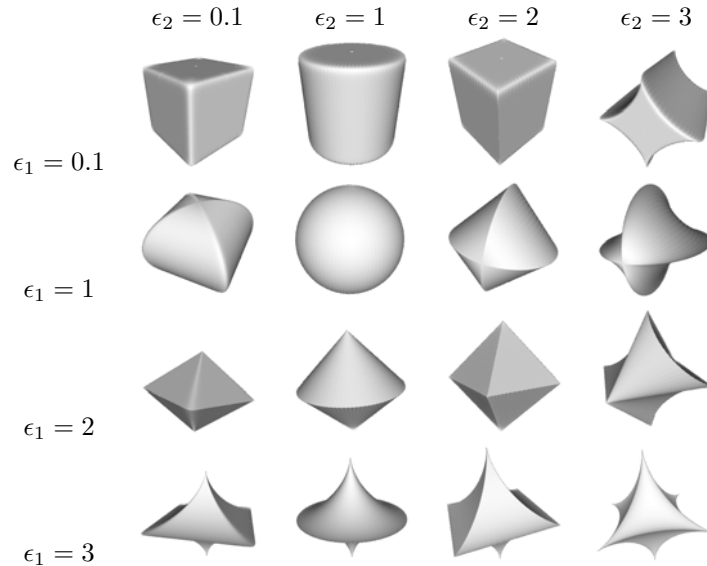


Figure 1: Superquadrics are compact parametric shapes with triaxis symmetry that are appropriate for modeling frequently encountered objects in most man-made environments.

2 METHOD

After capturing the color and depth images with the Kinect 2 camera and registering them to obtain a RGB-D colored image, the proposed method proceeds in two steps. In the first step, the objects in the scene are segmented and in the second one the moments of each object are computed by fitting a superquadric to the corresponding point cloud. Next, some details are given for each of these two steps. The approach is similar to that proposed in [2], and [12], but here the segmentation and approximation steps are explicitly separated and not integrated in the same process. The approach is also related to that presented in [5], but here the use of superquadrics provide more flexible approximations to the detected objects.

2.1 OBJECT SEGMENTATION

A myriad of methods for segmenting objects in a depth map into sets of points that are likely to belong to the same objects have been reported (see, for example, [11] and the references therein). In our case, the scene is segmented into individual objects using a classical region growing segmentation process [13] based on normal and curvature similarity. The algorithm, included in the PCL, starts computing the normal and the curvature for all points in the scene. Then, the unlabeled point with less curvature is selected and all its adjacent points with similar normal and curvature are added to the corresponding region. The similarity is evaluated using user-provided thresholds for each measure. As the region growing process

progresses, the new adjacent points are also considered as candidates to joint the current region. When the region can not be grown any more, the overall process is repeated with a new starting point until all the points in the scene are labeled. To remove noisy regions, patches with less than a given number of points (300 in our implementation) are not considered. The obtained segmentation is typically of good quality due to the integration of the information provided by the the normals and the curvature.

2.2 FITTING SUPERQUADRATICS

Superquadrics are a family of geometric shapes defined by formulas that resemble those of ellipsoids, but with arbitrary powers. The superquadrics are a versatile object modeling tool in computer graphics [1, 8]. Although a superquadric can represent different shapes, modeling is typically based on superellipsoids. The implicit equation of a superellipsoid is

$$\left[\left(\frac{x}{a} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{b} \right)^{\frac{2}{\epsilon_2}} \right]^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{c} \right)^{\frac{2}{\epsilon_1}} = 1, \quad (1)$$

where

- a, b, c are the scaling factors in the X, Y , and Z axis, respectively and, thus indicate the approximated size of the object in each dimension.
- ϵ_1 controls the shape of the superellipsoid in the plane containing the Z axis
- ϵ_2 controls the shape of the superellipsoid in the planes parallel to the $X - Y$ plane.



Figure 2: A scene used as an example of a cluttered pile of man-made objects (left), and the depth map acquired using a Kinect 2 sensor (right).

Figure 1 shows the shape versatility of the superellipsoids in terms of ϵ_1 and ϵ_2 .

Using the procedure described in [6] and implemented in [7], to fit a superellipsoid to a given point cloud, the point cloud is zero-centered and the principal axis of the cloud, computed with principal component analysis (PCA), are aligned with the $X - Y - Z$ axis. Then, ϵ_1 and ϵ_2 are set to 1 and a , b and c are set to 3 times the singularvalues obtained with the PCA [2, 20]. From this initial point a local optimization based on the Levenberg-Marquardt algorithm is used to obtain the final parameter estimation [4, 8, 9]. The overall process is fast and provides an approximations which, in general, captures the position, the orientation and the overall shape of the input point clouds. Thus, the proposed method can be seen as a fast alternative to more complex shape approximation approaches [10, 14].

3 EXPERIMENTS

To validate the proposed approach we use a scene with objects with different shapes. Color and depth images are obtained with a Kinect 2 camera. The color image is shown in Figs. 2-left. The color and depth images are registered to obtain the colored depth map shown in Fig. 2-right.

Then, the segmentation process described in Section 2.1 is applied and 10 regions with similar normals and curvature are identified. These regions are shown in Fig. 3 in different colors. The points not assigned to any region (i.e., points forming regions with less than 300 points) appear in red. It can be seen that the largest region corresponds

to the background of the scene. Note how the cylinder and the background have similar colors, but they are correctly segmented due to the use of normal and curvature information in the segmentation process. However, the use of this information can cause confusion in some cases. For instance, the outer and the inner parts of the cylinder are segmented as two regions due to their difference in curvature. Such regions would probably be merged if considering color information. Despite these spurious effects the use of the normal and curvature information is beneficial in general. Figure 3-right shows a perspective view of the segmented scene.

Each patch is now approximated by a superellipsoid using the procedure described in Section 2.2. Figure 4 shows the point patches and the corresponding approximation. All the approximations are close enough to the corresponding patch. Probably the less accurate approximation is that of the cylinder (patch 3). The overall shape of the cylinder is properly captured but the superellipsoid is too large. This is due to a segmentation problem which includes spurious points in the cylinder region. During the approximation process the superellipsoid is enlarged with the aim of fitting it to these points too.

The parameters for each approximation are given in Table 1. Parameters ϵ_1 and ϵ_2 control the shape of the ellipsoid and a , b , and c give information about its size along the coordinate axis. Vectors \mathbf{p}_c and \mathbf{p}_s are the position and the roll, pitch, and yaw (in radians) of the principal axis considering only the point cloud and using the superellipsoid approximation, respectively. It can be

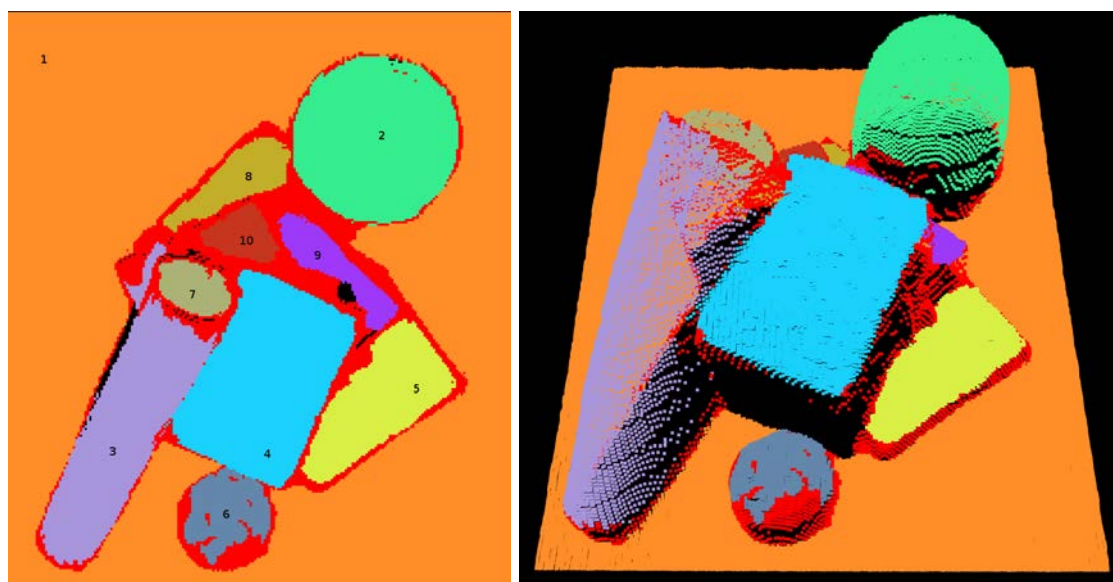


Figure 3: Two views of the segmented scene. Red pixels are points not included in any object.

Table 1: Parameters defining the superellipsoid approximating each patch. The patch numbers are indicated in Fig. 3-left.

Patch	Points	ϵ_1	ϵ_2	a	b	c	\mathbf{P}_c	x, y, z r, p, y	\mathbf{P}_s	x, y, z r, p, y
1	31397	0.81	1.67	461.51	399.53	9.46	-314.10,-1068.86,1100.40 -2.03,-0.05,0	-326.58,-1048.26,1098.64 -2.31,-0.05,-3.14		
2	4050	0.90	0.96	105.92	104.82	108.32	-473.94,-1261.67,947.95 -1.13,0.07,0.11	-476.96,-1262.78,1014.19 -2.04,0.30,0.31		
3	3605	0.63	0.70	105.08	336.78	48.54	-147.63,-891.56,886.01 -0.47,1.16,0.40	-185.10,-976.50,921.18 1.19,1.18,1.82		
4	4364	0.50	0.86	110.47	85.55	6.10	-330.84,-950.57,889.33 -1.12,0.05,0.02	-334.96,-948.04,891.69 -1.39,0.12,0.07		
5	2164	1.26	1.50	148.04	67.83	4.41	-477.98,-924.34,1076.57 -0.82,0.04,0.01	-479.65,-925.84,1076.62 -0.81,0.05,0.01		
6	984	0.99	0.98	74.97	54.14	47.63	-280.90,-780.31,1022.31 -1.14,0.15,-0.66	-280.88,-772.45,1057.76 -1.40,0.29,-0.86		
7	552	0.24	1.04	42.81	33.08	9.11	-238.86,-1069.44,840.98 -0.56,0.07,-2.61	-238.96,-1068.08,835.91 -0.65,0.05,-2.63		
8	889	0.59	1.46	30.78	91.25	5.38	-294.20,-1208.58,1074.44 -0.57,0.05,0	-292.23,-1207.21,1075.08 2.16,0.04,3.14		
9	796	1.26	1.45	30.75	130.11	5.94	-421.48,-1096.88,1002.21 -0.74,-0.05,3.08	-422.74,-1094.77,1002.16 2.32,-0.05,0.05		
10	528	0.57	1.40	33.29	42.17	6.19	-301.88,-1137.26,963.06 -0.21,-0.01,3.08	-302.64,-1133.94,964.04 1.74,-0.02,0.05		

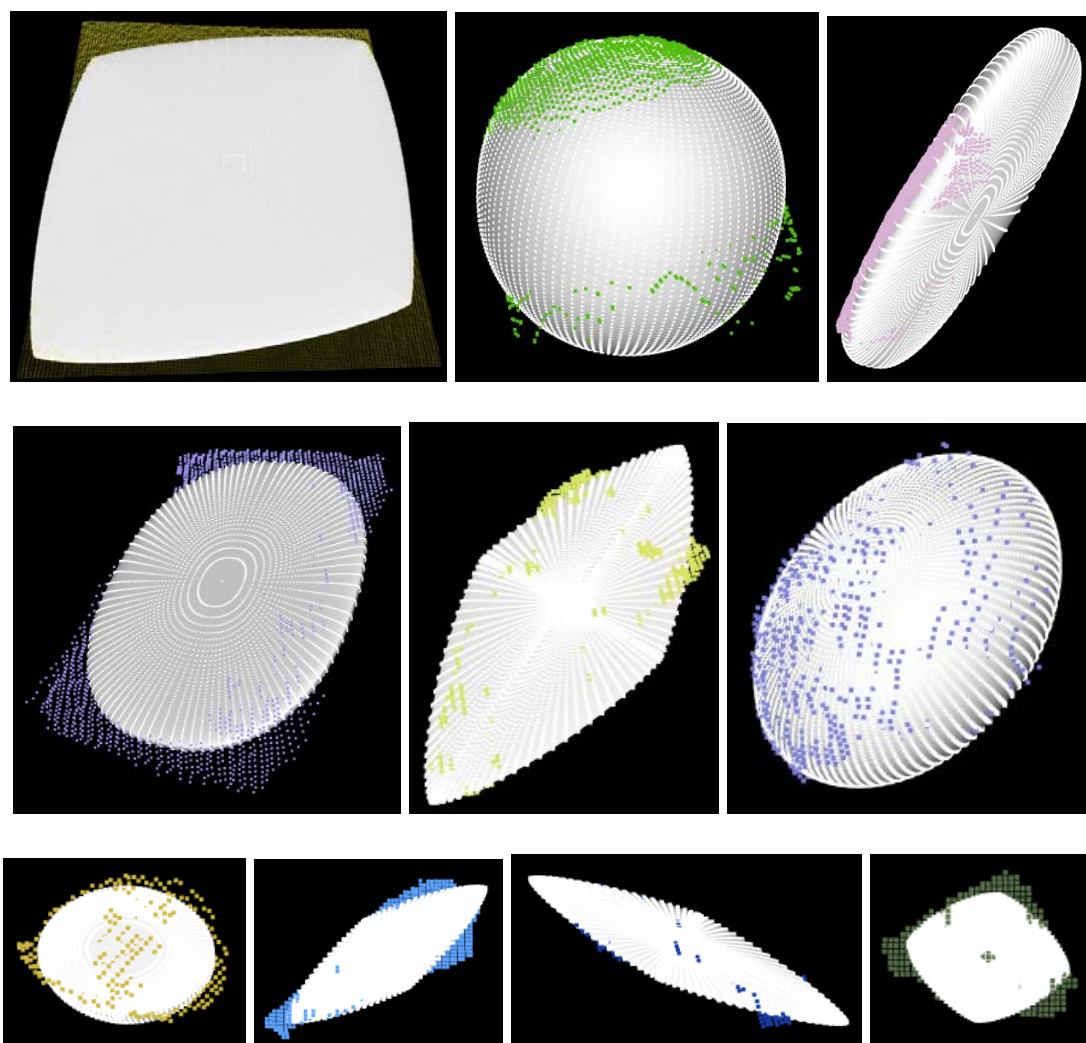


Figure 4: The 10 patches obtained from the segmentation and their corresponding approximation with a superellipsoid.

seen that the approximation with a superellipsoid significantly modifies these parameters. This is clear, for instance, in the case of the ball (patch 2) where the centroid obtained into account only the point data is closer to the camera than the one computed from the superellipsoid. This second one correctly approximates the actual center of the ball. Most of the patches are planar (corresponding to the scene background or the books covers). Such patches are approximated with quasi planar superellipsoids, i.e., ellipsoids with a c parameter significantly smaller than the a and b parameters. The values of such parameters are similar to each other in the approximation of the two balls (patches 2 and 6).

Figure 5 shows the representation of the input scene using the obtained superellipsoids. Note that using the superellipsoids we can summarize an initial cloud with more than half a million points with merely 110 parameters while still re-

taining the relevant information to select candidate grasps.

4 CONCLUSIONS

This paper presents preliminary work towards the determination of good grasping points from point clouds obtained from a Kinect 2 camera. The steps performed so far include the segmentation of the point clouds with a region growing algorithm and the approximation of the segmented objects with superellipsoids. This representation is very compact and provides more accurate information on the principal axis of the object than the one directly obtained from the point cloud.

This preliminary work can be extended in several ways. First, sub-pixel depth estimation can be obtained integrating the color and the depth informations. Moreover, in our approach, segmentation and geometric primitive extraction is car-

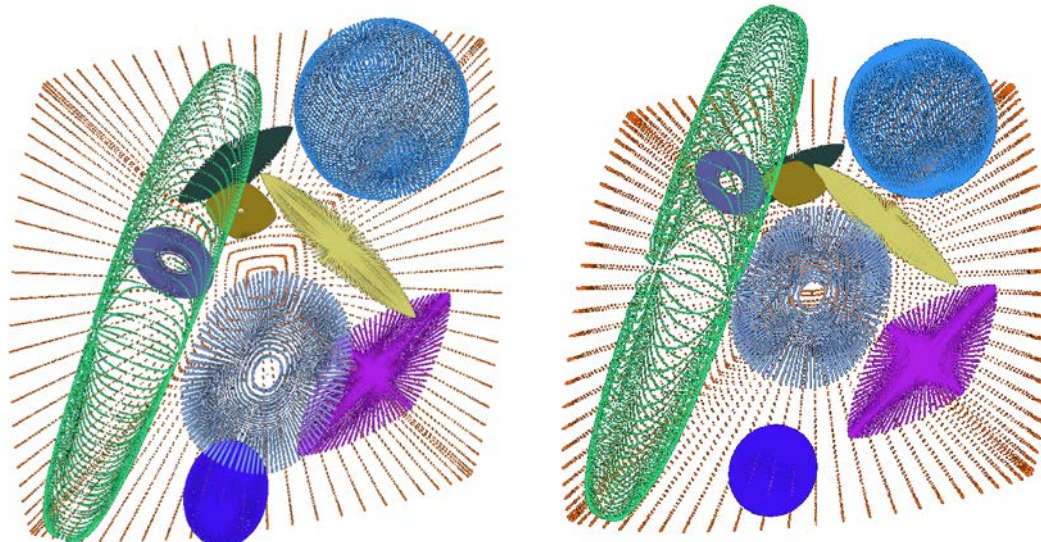


Figure 5: Two views of the final scene representation.

ried out independently of each other. However, some authors have proposed methods that concurrently perform both operations [2, 12]. In our case, this idea seem worth exploring. For instance, in our approach a cube-like object would be approximated with several superquadrics, one for each detected face. In a integrated approach, a single superquadric could be generated to approximate the object. Another point deserving further attention is the real time implementation of proposed approach which will probably require the use of specialized hardware, like field programmable gate arrays (FPGAs). Finally, higher quality models can be obtained integrating the information from different cameras. Such high quality models would facilitate the determination of the grasping points following the approach presented in [5].

Acknowledgements

This work has been partially supported by by the Spanish Ministry of Economy and Competitiveness under projects DPI2014-57220-C2-2-P and DPI2014-57746-C3.

References

- [1] Barr, A. H., “Superquadrics and angle-preserving transformations,” *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.
- [2] Biegelbauer, G. and Vincze, M., “Efficient 3D object detection by fitting superquadrics to range image data for robot’s object manipulation,” *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1086-1091, 2007.
- [3] Borst, C., Fischer, M., and Hirzinger, G., “A fast and robust grasp planner for arbitrary 3D objects,” *IEEE International Conference on Robotics and Automation (ICRA)*, 1999.
- [4] Chevalier, L., Jaillet, F., and Baskurt, A., “Segmentation and superquadric modeling of 3D objects,” *Winter School of Computer Graphics*, 2003.
- [5] Fornas, D., Sales, J., Peñalvera, A., Pérez, J., Fernández, J. J., Marín, R., and Sanz, P. J., “Fitting primitive shapes in point clouds: a practical approach to improve autonomous underwater grasp specification of unknown objects,” *Journal of Experimental & Theoretical Artificial Intelligence*, to appear, 2016.
- [6] A.-E. Ichim, “Superquadrics”, PCL Toyota Code Sprint 2.0, 2013.
- [7] A.-E. Ichim, <https://github.com/aichim/superquadrics>, 2013.
- [8] Jaklic, A., Leonardis, A., and Solina, F., “Segmentation and Recovery of Superquadrics,” *Computational imaging and vision*, Vol. 20, Kluwer, Dordrecht, 2000.
- [9] Katsoulas, D., Bastidas C. C., and Kosmopoulos, D. I., “Superquadric segmentation in range images via fusion of region and boundary information,” *IEEE Transactions on Pattern Analysis Machine Intelligence*, 30(5):781–795, 2008.
- [10] Mörwalda, T., Balzerb, J., Vinczea, M., “Modeling connected regions in arbitrary planar point clouds by robust B-spline approxi-

- mation,” *Robotics and Autonomous Systems*, 76:141–151, 2016.
- [11] Nguyen, A., Le, B., “3D point cloud segmentation: A survey,” *IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pages 225-230, 2013.
- [12] Pascoal, R., Santos, V., Premebida, C., and Nunes, U., “Simultaneous Segmentation and Superquadrics Fitting in Laser-Range Data,” *IEEE Transactions on Vehicular Technology*, 64(2):441-452, 2015.
- [13] Pratt, W. K., “Digital Image Processing 4th Edition,” John Wiley & Sons, Inc., Los Altos, California, 2007.
- [14] Richtsfeld, A., Mörwald, T., Prankl, J., Zillich, M., and Vincze, M., “Segmentation of Unknown Objects in Indoor Environments,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4791-4796, 2012.
- [15] Rosales, C., Porta, J. M., Suarez, R., Ros, L., “Finding All Valid Hand Configurations for a given Precision Graps,” *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1634-1640, 2008.
- [16] Rusu, R. B., and Cousins, S., “3D is here: Point Cloud Library (PCL),” *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.
- [17] Schwarz, M., Schulz, H., and Behnke, S., “RGB-D Object Recognition and Pose Estimation based on Pre-trained Convolutional Neural Network Features,” *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1329-1335, 2015.
- [18] Taubin, G., and Cooper, D. B., “Object Recognition Based on Moments (or Algebraic) Invariants,” *Geometric Invariance in Computer Vision*, J.L. Mundy and A. Zisserman, eds., MIT Press, pp. 375-397, 1992.
- [19] Zennaro, S., Munaro, M., Milani, S., Zanuttigh, P., Bernardi, A., Ghidoni, S., Menegatti, E., “Performance evaluation of the 1st and 2nd generation Kinect for multimedia applications,” *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1-6. 2015.
- [20] Xiao, Y. and Siebert, J., “Building superquadric men from 3d whole-body scan data,” *IEEE Chapter Conference on Applied Cybernetics*, pages 82–88, 2005.