3D Scene Reconstruction Using Kinect

Marco Morana

Abstract The issue of the automatic reconstruction of 3D scenes has been addressed in several chapters over the last few years. Many of them describe techniques for processing stereo vision or range images captured by high quality range sensors. However, due to the high price of such input devices, most of the methods proposed in the literature are not suitable for real-world scenarios. This chapter proposes a method designed to reconstruct 3D scenes perceived by means of a cheap device, namely the Kinect sensor. The scene is efficiently represented as a composition of superquadric shapes so as to obtain a compact description of environment, however complex it may be. The approach proposed here is intended to be used as a novel processing module of a well-established cognitive architecture for artificial vision. Experimental tests have been performed on real images and the results look very promising.

1 Introduction

Over the last 40 years, the issue of automatically recognizing real-world objects has been investigated by a considerable body of research related to different fields, from computer vision to neuroscience. The techniques proposed therein can be roughly classified as those recognizing the objects contained in a scene in a 2D or 3D space. Both 2D and 3D object recognition still present challenges for the computer science community since the same object usually looks very different according to its orientation, scale and more generally to the acquisition conditions.

A further distinction can be made between "full object recognition" and "recognition by parts" approaches. In many cases the latter is preferred since a complex object can be described as a combination of simpler primitives which can be related to each other by logical relations (e.g., *above*, *below*, *larger*, *smaller* and so on).

M. Morana (✉)
University of Palermo, Viale delle Scienze, Edificio 6, 90128 Palermo, Italy
e-mail: marco.morana@unipa.it

S. Gaglio and G. Lo Re (eds.), *Advances onto the Internet of Things*,
Advances in Intelligent Systems and Computing 260, DOI: 10.1007/978-3-319-03992-3_13,

In this chapter, we describe a framework for efficiently representing a 3D scene as a combination of superquadric curves. In particular, the object is perceived by means of a cheap device containing both an RGB camera and a depth sensor, namely the Microsoft Kinect. A volumetric analysis is then performed to discard noisy data and the object is reconstructed by estimating a set of best-fitting superquadrics.

The chapter is organized as follows: related works are outlined in Sect. 2, whilst the system architecture proposed here is described in Sect. 3. Experimental results are detailed in Sect. 4, and conclusions are discussed in Sect. 5.

2 Related Work

A mutual relationship exists between scene reconstruction and object recognition processes. The reason for this is that, in order to reconstruct a scene it is useful to break the scene down into objects. Then, once a description of the scene has been provided, it is possible to recognize the observed objects by classifying their descriptors.

Several systems for 3D object representation have been proposed over the last few years. The main challenge of such approaches is to obtain satisfactory results not only in a controlled testing environment, but also in complex scenarios with unconstrained conditions, e.g., a home environment or an office. In many cases, range images, i.e., 2D images in which each pixel contains the distance between the sensor and a point in the scene, are preferred to the RGB ones since they generally provide a better discriminable data representation.

Since range images are more robust in the face of changes in environment conditions, a number of works have focused on how they should be processed.

In [13], an approach for the direct recovery of a set of volumetric models, i.e., superquadrics, from unsegmented range data is presented. The method is divided into two stages: model-recovery and model-selection. During the first stage, several seeds are placed at random points in the input image, and for each seed, a model is iteratively built and allowed to grow. Finally, those models which produce the simplest and most accurate approximation of the input data are selected.

A technique for part-level object recognition using superquadrics is presented in [12]. The system is based on interpretation trees [10] and can handle flexible articulated objects, i.e., human figurines, that cannot be perfectly modeled by superquadrics.

In [15], a framework is described for extracting some 3D primitives (i.e., spheres, cylinders, cones) from range data captured by a laser scanner.

Several systems provide good results, although high quality range sensors are needed to obtain high resolution input images. Since range sensors are usually very expensive, most of the methods proposed so far have not been suitable for extensive use in real-world scenarios. For this reason, our proposal involves the use of a cheap device containing both an RGB camera and a depth sensor.

The method proposed here is intended to be used as a novel processing module of the framework presented in [4, 5]. In their work, the authors describe a cognitive

architecture for an artificial vision system, in which an effective internal representation of the environment is built up by means of processes defined over a suitable intermediate level, the *conceptual level*, that acts between the sensory data, the *sub-symbolic level* and the linguistic *symbolic level*. In particular, the conceptual level is characterized by a conceptual space whose dimensions are the parameters of the 3D geometric primitives, i.e., superquadrics, which constitute the scene. The aim of this work is to provide a more efficient technique for reconstructing 3D objects by means of the Kinect sensor.

Microsoft Kinect is based on the hardware reference design and the structured-light decoding chip provided by PrimeSense, an Israeli company which also provides a framework, OpenNI [16], that supplies a set of APIs to be implemented by sensor devices and middleware components.

The core of the Kinect is represented by the vision system composed of an RGB camera with VGA standard resolution (i.e., 640 $\times$ 480 pixels), an IR projector that shines a grid of infrared dots over the scene and an IR camera that captures the infrared light. The factory calibration of the Kinect makes it possible to establish the exact position of each projected dot against a surface at a known distance from the camera. The deformation of this dot pattern against the scene is captured to derive depth images of the observed scene, and capture the objects' position in a three-dimensional space.

Even though Kinect has only been on the market for a couple of years, it has attracted the attention of a number of researchers, thanks to the availability of open-source and multi-platform libraries that reduce the cost of developing new algorithms. A survey of the sensor and corresponding libraries is presented in [3, 11]. In [1], an approach based on RANSAC (Random Sample Consensus) [9], an algorithm for robustly fitting models in the presence of many data outliers, is described. The authors proposed a solution for 3D object localization using superquadrics to model image data captured by the Kinect. Because it is easy to use, the Kinect sensor has also been successfully adopted as an input device for gesture [14] or activity [6] recognition systems in ambient intelligence scenarios.

3 System Overview

In this section a description of the system is given, explaining both the basis of superquadric shapes and the reconstruction technique proposed here.

3.1 Superquadrics

The term *superquadrics* was first used by [2] to define a family of geometric shapes that includes superellipsoids, superhyperboloids of one piece, superhyperboloids of two pieces and Supertoroids.

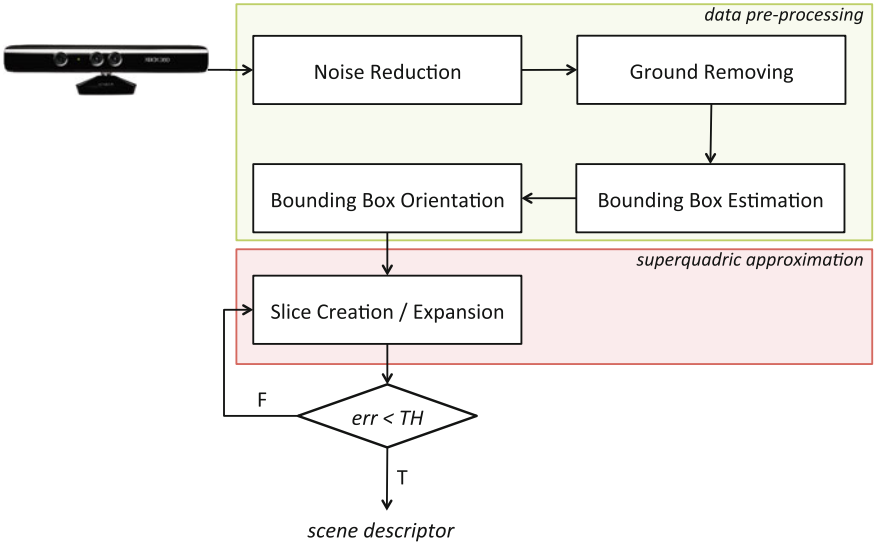

Fig. 1 System overview

The explicit form of a superquadric is given by the equation:

$$S_p(\eta, \omega) = \begin{bmatrix} x(\mathbf{p};\ \eta,\ \omega) \\ y(\mathbf{p};\ \eta,\ \omega) \\ z(\mathbf{p};\ \eta,\ \omega) \end{bmatrix} = \begin{bmatrix} a_1 \cos(\eta)^{\varepsilon_1} \cos(\omega)^{\varepsilon_2} \\ a_2 \cos(\eta)^{\varepsilon_1} \sin(\omega)^{\varepsilon_2} \\ a_3 \sin(\eta)^{\varepsilon_1} \end{bmatrix} \tag{1}$$

where $-\pi/2 \leq \eta \leq \pi/2$ and $-\pi \leq \omega \leq \pi$.

The elements of the vector $\mathbf{p} = (a_1,\ a_2,\ a_3,\ \varepsilon_1,\ \varepsilon_2)$ are the parameters of the superquadric. In particular, a_1, a_2, a_3 represent the size of the model along the **X**, **Y**, **Z** axes, and ε_1, ε_2 control the shape of the model. More specifically, ε_1 is the squareness parameter in the north-south directio , while ε_2 is the squareness parameter in the east-west direction (see Fig. 2).

The inside-outside equation of the superquadric in implicit form is:

$$F(x,\ y,\ z) = \left[\left(\frac{x}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right]^{\frac{\varepsilon_2}{\varepsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\varepsilon_1}} \tag{2}$$

where $F(x,\ y,\ z)$ assumes a value equal to 1 when the point $(x,\ y,\ z)$ is a superquadric boundary point, a value less than 1 when it is an inside point, and a value greater than 1 when it is an outside point.

In order to model a superquadric in a general position, six additional parameters are needed. In particular, $p_x,\ p_y,\ p_z$ define the translation of the model relative to

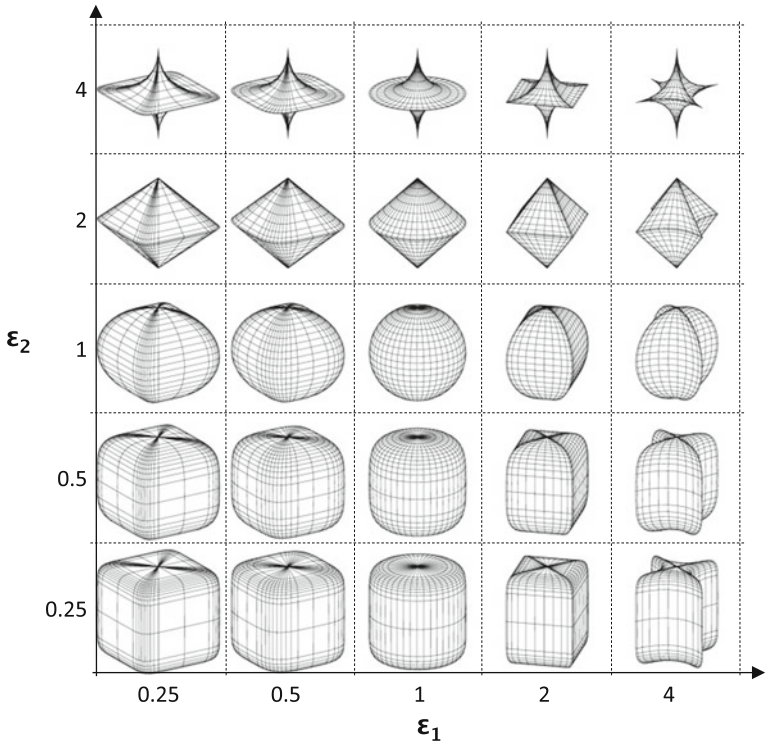

Fig. 2 Shapes obtained with $\varepsilon_1,\ \varepsilon_2$ in the range $[0,\ 4]$

the origin of the coordinate system, while the orientation in space is expressed by means of the angles $\phi,\ \theta,\ \psi$.

Thus, the model parameter vector $\mathbf{p}$ in the general position is:

$$\mathbf{p} = \left(a_1,\ a_2,\ a_3,\ \varepsilon_1,\ \varepsilon_2,\ p_x,\ p_y,\ p_z, \phi,\ \theta,\ \psi\right) \tag{3}$$

3.2 Scene Reconstruction

The method proposed in this chapter aims to reconstruct 3D scenes captured by the Kinect as a composition of some superquadric shapes. As previously discussed, research in the literature has addressed this problem by processing the images made by traditional range cameras or stereo vision systems. Here, in order to obtain a more detailed data representation, we directly process the 3D point cloud captured by the Kinect.

In order to correctly approximate the object some data pre-processing is required. Firstly, the whole set of 3D points (Fig. 3c) is analyzed to reduce the noise related to

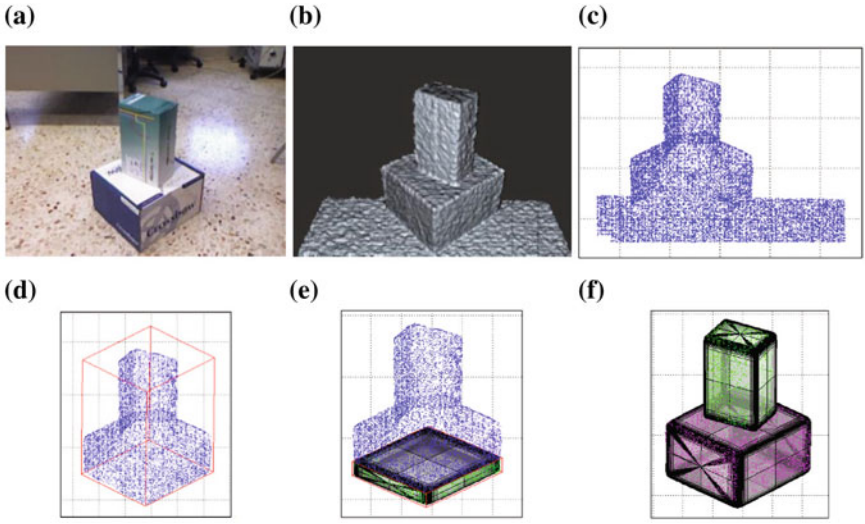

Fig. 3 Example of some processing steps. RGB image (**a**), depth image (**b**), point cloud (**c**), ground removing and bounding box (**d**), superquadrics obtained from the initial slice (**e**), optimal approximating superquadrics (**f**)

the acquisition process, i.e., points not belonging to the object or to the scene. The noise reduction method computes the distance between a couple of points, discarding those points whose distance is above a given threshold. The maximum distance is dynamically computed according to the mean distance measured for the considered point cloud.

Once the set of points has been filtered, a ground removing algorithm is applied to separate the object from the plain it lies in. The algorithm, based on RANSAC, computes the plane defined by 3 randomly chosen points and evaluates the number of inliers for that plane. This process is repeated for a certain number of iterations and the best plane, that is the plane with the greater number of inliers, is selected as *ground*.

Next, an overall bounding box BB_O is estimated for the whole set of points (Fig. 3d) and, in order to correctly break up the object into slices, the point cloud is rotated to the angle needed to arrange the bounding box parallel to the 3D axes.

As shown in Fig. 1, the superquadric approximation process is based on an iterative procedure for the creation and expansion of slices of 3D points.

The creation of a slice consists in the selection of a set of 3D points in a randomly chosen direction. For example, a slice of height H in the z-direction is created by selecting the (x, y, z) points of the cloud in the range $z_{\min} \leq z \leq z_{\max}$, where $z_{\max} - z_{\min} = H$.

In order to find the superquadrics that best approximates the point cloud contained in each slice, both the scale parameters a_1, a_2, a_3 and the form factors ε_1, ε_2 need to be defined. In particular, the size of the superquadric is estimated according

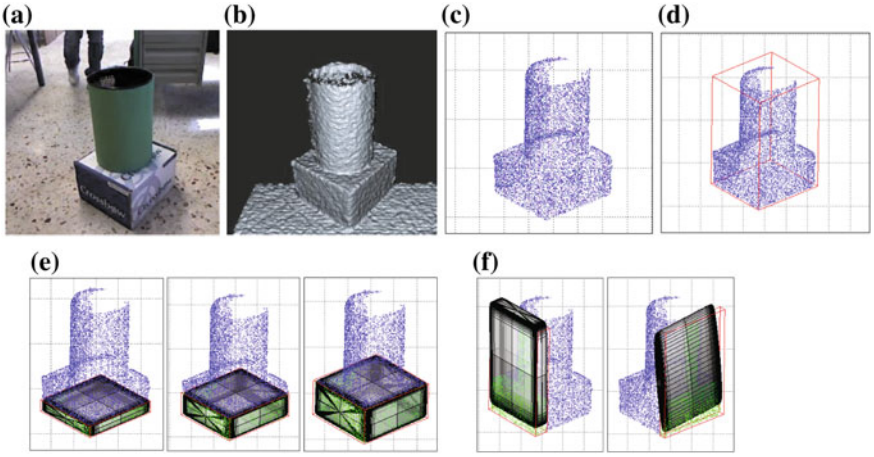

Fig. 4 Example of the slice creation and expansion process. RGB image (**a**), depth image (**b**), point cloud (**c**), ground removing and bounding box (**d**), superquadrics obtained from a correct slice selection (**e**), superquadrics obtained from an incorrect slice selection (**f**)

to the dimensions of the minimal bounding box BB that fits the set of 3D points contained in a slice, that is $a_1 = BB_x/2$, $a_2 = BB_y/2$, $a_3 = BB_z/2$. The form factors are computed by applying the RANSAC algorithm to search for the couple $(\varepsilon_1, \varepsilon_2)$ in the range $[0, 1]$ that best fits the input points. The remaining parameters $(p_x, p_y, p_z, \phi, \theta, \psi)$ are computed according to the position and orientation of BB. Note that the dimensions of BB are dependent on the number of 3D points effectively discovered in each slice region. In fact, since the Kinect is able to capture only those points belonging to the object surfaces, it usually happens that no points are selected in a particular direction.

Once the superquadric has been computed, the fitting error, i.e., a measure of how well the current model fits the points of the slice, is computed according to the least-squares minimization of the superquadric inside-outside function (Eq. 2) proposed in [17].

During the expansion step the size of the slice is increased in the chosen direction, e.g., the z-direction in the example given above. Then the fitting error e_i at the step i is compared with a threshold TH and the current slice is expanded until $e_i > TH$.

Once a slice can not be expanded any further, the method continues the processing of the remaining point cloud by iterating the slice creation-expansion steps until the whole scene has been analyzed.

Figure 4 shows the processing steps involved in the approximation of an object composed of a box and a cylinder. In particular, the images in Fig. 4e show the superquadrics obtained from the selection of a correct slice, while an example of slice selection along two incorrect directions is shown in Fig. 4f.

Once the object has been approximated, it can be fully described by the whole set of parameters of the approximating superquadrics.

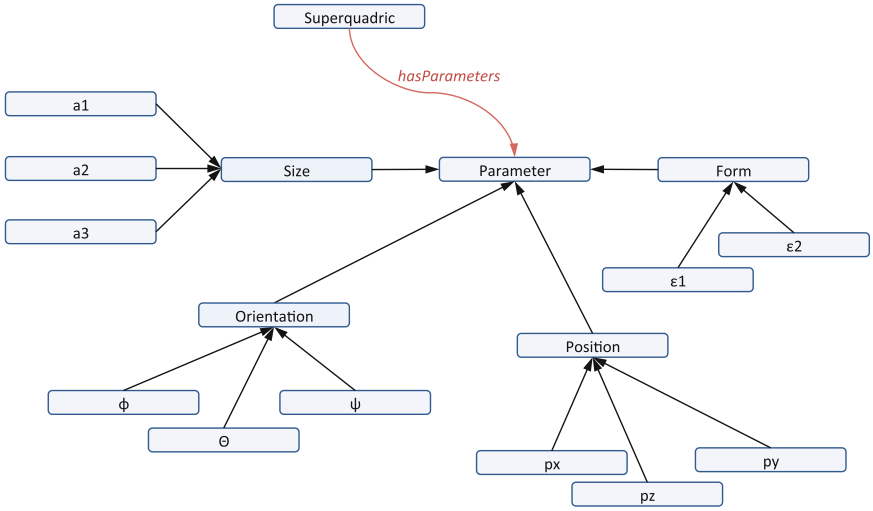

Fig. 5 The ontology representing the superquadrics

Information gathered through the reconstruction process is represented by means of the ontology shown in Fig. 5. As discussed above, the **Superquadric** shape is defined by a set of **Parameters** that capture properties related to the size, form, orientation and position of the curve, i.e, the object. Thus, the **Size** parameters a_1, a_2, a_3, the **Form** parameters ε_1, ε_2, the **Orientation** parameters ϕ, θ, ψ and the **Position** parameters p_x, p_y, p_z fully describe the **Superquadric** in the 3-D space.

4 Experimental Results

The proposed architecture has been designed to address a specific application scenario involving the management of indoor environments, e.g., offices or homes [8]. The main characteristic of such environments is that their interior design is usually based on a number of objects (e.g., chairs, desks, bookcases) that can be successfully represented as a composition of simple shapes (e.g., parallelepipeds, spheres, cylinders). In the AmI architecture adopted, a Wireless Sensor and Actuator Network (WSAN), whose nodes are equipped with off-the-shelf sensors (i.e., outdoor temperature, relative humidity, ambient light exposure and noise level) [7] is used to monitor the whole environment, while the Kinect sensor is used to detect specific objects placed within the office.

In order to evaluate the accuracy of the proposed scene reconstruction module in a real world scenario, several tests were performed on data captured by means of a Kinect device. In particular, we wanted to understand how some objects' properties

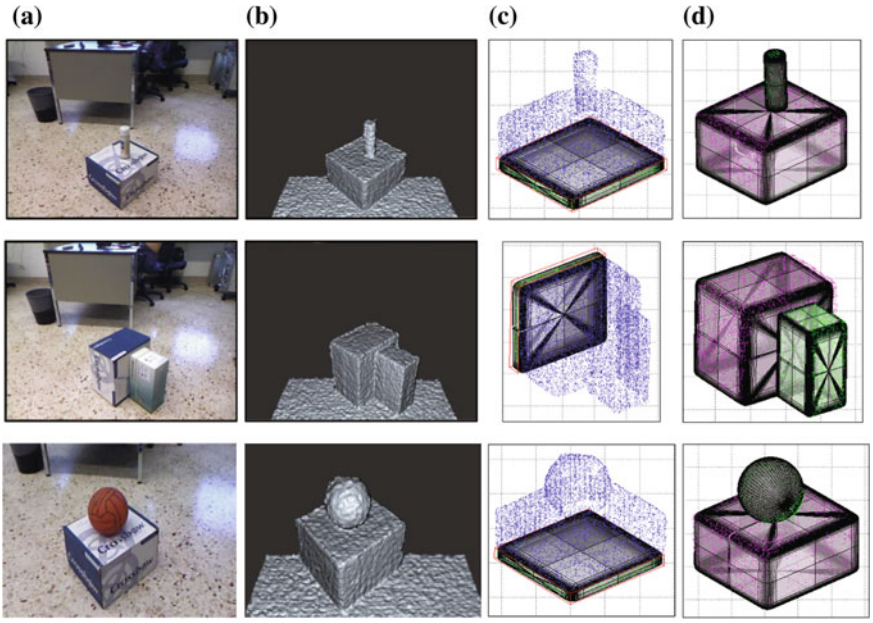

Fig. 6 Superquadric approximation of three simple scenes consisting of a single composite object. RGB image (**a**), depth image (**b**), superquadrics obtained from initial slice selection (**c**) and (**d**), optimal approximating superquadrics

(e.g., size, position, material) would eventually affect the overall performance of the proposed method.

Tests were conducted on 3D objects that can be broken down into different configurations of adjacent cubes, parallelepipeds, cylinders or spheres. These basic shapes are obtained by limiting the possible values of ε_1, ε_2, so the same approach could thus easily be extended to more complex shapes by considering different values of the ε_1, ε_2 parameters.

Some significant examples of scene reconstructions are shown in Figs. 6 and 7. The set of tests shown in Fig. 6 is oriented to observe how the system deals with objects that can be approximated with two simple superquadric shapes. The first row shows some images related to the reconstruction of a scene consisting of a spray placed above a box. This test serves to evaluate the ability of the proposed approach in approximating small noisy objects, such as the spray. The second row shows the reconstruction of two adjacent boxes. This kind of test was performed to evaluate how efficiently partial occlusions are managed. The third row shows the reconstruction of a scene consisting of a ball placed above a box. This test allowed us to demonstrate that symmetric and partially occluded objects, i.e., the ball, can be successfully processed.

The reconstruction of three more complex scenes is shown in Fig. 7. The difficulty associated with these scenes is mainly represented by the limited amount of free space

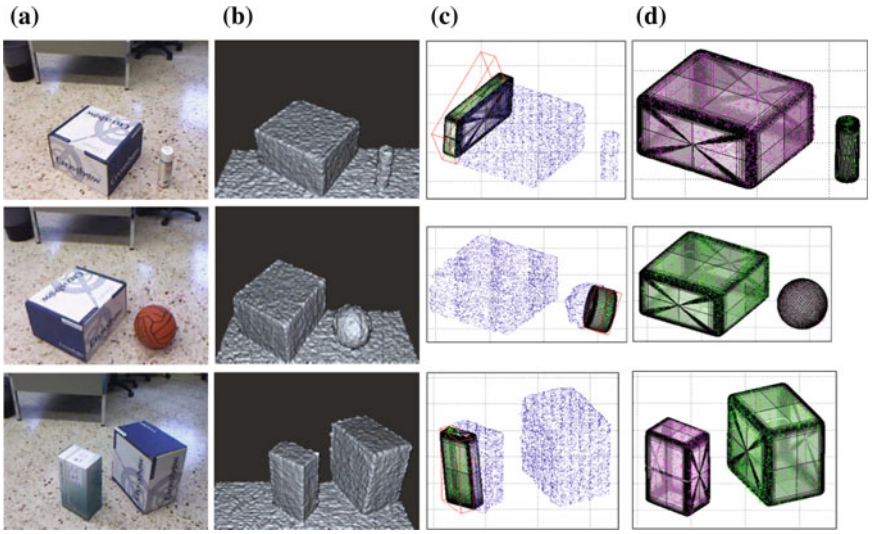

Fig. 7 Superquadric approximation of three more complex scenes consisting of two distinct objects. RGB image (**a**), depth image (**b**), superquadrics obtained from initial slice selection (**c**) and (**d**), optimal approximating superquadrics

between the three pairs of captured objects. For example, the top row shows that the smaller object (i.e., the spray) is detected and correctly approximated even though it is close to a bigger object characterized by a larger number of points.

The method proposed was been tested on about 30 scenes with different levels of complexity. For each scene, the whole process was run 10 times, obtaining an average reconstruction rate of 84 %.

The prototype was implemented connecting the Kinect to a personal computer (i.e., 2.5 GHz dual-core Intel Core i5, 4 GB of RAM and Unix OS) running MATLAB. The average scene reconstruction takes about 1–2 min.

From the analysis of the experimental results it emerges that some constraints need to be satisfied during the acquisition process. In particular, we noticed that three sides of the object should always be visible from the Kinect's point-of-view. This requirements has to be met to correctly drive the bounding box estimation process. Otherwise, it would not be possible to determine the scale parameters and consequently the form factors.

Moreover, some objects cannot be correctly captured by the Kinect because of to the material they are made of (see Fig. 8). For example, reflecting objects cause the IR ray to be reflected and lost, whilst transparent objects are not-correctly reconstructed since the ray is distorted when passing through them.

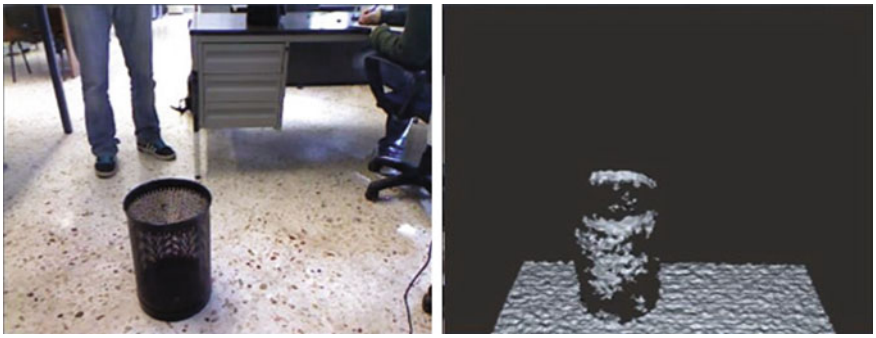

Fig. 8 Example of a misreconstructed object

5 Conclusions

This chapter describes a system for the automatic reconstruction of 3D objects captured by means of the Kinect sensor.

As compared to other solutions for object reconstruction from range images or stereo vision systems, the goal here was to demonstrate that composite objects can be efficiently reconstructed and represented by using inexpensive devices.

The experimental results demonstrate that the quality of the images provided by the Kinect is good enough to obtain satisfactory results, even under partially constrained conditions.

We are already working on improving the decomposition module in order to be able to reconstruct a greater set of composite objects, and that is, to consider a wider range of superquadric shapes. Moreover, once we have tested the effectiveness of our approach, we are planning a more efficient implementation of the prototype to speed up the reconstruction time.

Acknowledgments This work has been partially supported by the PO FESR 2007/2013 grant G73F11000130004 funding the SmartBuildings project.

References

1. Afanasyev, I., Biasi, N., Baglivo, L., Cecco, M.D.: 3D object localization using superquadric models with a kinect sensor. Technical report: Mechatronics Department, University of Trento, Italy (2011). http://www.ing.unitn.it/afanasye
2. Barr, A.: Superquadrics and angle-preserving transformations. Comput. Graph. Appl. IEEE **1**(1), 11–23 (1981). doi:10.1109/MCG.1981.1673799
3. Borenstein, G.: Making Things See: 3D Vision with Kinect, Processing, Arduino, and MakerBot. Make: Books. O'Reilly Media Inc., Sebastopol (2012)
4. Chella, A., Frixione, M., Gaglio, S.: A cognitive architecture for artificial vision. Artif. Intell. **89**(1–2), 73–111 (1997). doi:10.1016/S0004-3702(96),00039-2

5. Chella, A., Frixione, M., Gaglio, S.: Understanding dynamic scenes. Artif. Intell. **123**(1–2), 89–132 (2000). doi:10.1016/S0004-3702(00),00048-5
6. Cottone, P., Lo Re, G., Maida, G., Morana, M.: Motion sensors for activity recognition in an ambient-intelligence scenario. In: IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 646–651 (2013). doi:10.1109/PerComW.2013.6529573
7. De Paola, A., Gaglio, S., Lo Re, G., Ortolani, M.: Sensor9k : a testbed for designing and experimenting with wsn-based ambient intelligence applications. Pervasive Mob. Comput. **8**(3), 448–466 (2012). http://dx.doi.org/10.1016/j.pmcj.2011.02.006
8. De Paola, A., Lo Re, G., Morana, M., Ortolani, M.: An intelligent system for energy efficiency in a complex of buildings. In: Sustainable Internet and ICT for Sustainability (SustainIT), pp. 1–5 (2012)
9. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981). doi:10.1145/358669.358692. http://doi.acm.org/10.1145/358669.358692
10. Grimson, W.E.L.: Object Recognition by Computer: The Role of Geometric Constraints. MIT Press, Cambridge (1990)
11. Kean, S., Hall, J., Perry, P.: Meet the kinect: An Introduction to Programming Natural User Interfaces, 1st edn. Apress, CA (2011)
12. Krivic, J., Solina, F.: Part-level object recognition using superquadrics. Comput. Vision Image Underst. **95**(1), 105–126 (2004). doi:10.1016/j.cviu.2003.11.002
13. Leonardis, A., Jaklic, A., Solina, F.: Superquadrics for segmenting and modeling range data. IEEE Trans. Pattern Anal. Mach. Intell. **19**(11), 1289–1295 (1997). doi:10.1109/34.632988
14. Lo Re, G., Morana, M., Ortolani, M.: Improving user experience via motion sensors in an ambient intelligence scenario. In: Pervasive and Embedded Computing and Communication Systems (PECCS), pp. 29–34 (2013)
15. Marshall, D., Lukacs, G., Martin, R.: Robust segmentation of primitives from range data in the presence of geometric degeneracy. IEEE Trans. Pattern Anal. Mach. Intell. **23**(3), 304–314 (2001). doi:10.1109/34.910883. http://dx.doi.org/10.1109/34.910883
16. PrimeSense: Openni. http://www.openni.org/
17. Solina, F., Bajcsy, R.: Recovery of parametric models from range images: the case for superquadrics with global deformations. IEEE Trans. Pattern Anal. Mach. Intell. **12**(2), 131–147 (1990). doi:10.1109/34.44401