IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 30, 2020, accepted November 24, 2020, date of publication December 1, 2020,
date of current version January 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3041584

Learning to Predict Superquadric Parameters From Depth Images With Explicit and Implicit Supervision

TIM OBLAK[1,2], JAKA ŠIRCELJ[1,2], (Member, IEEE), VITOMIR ŠTRUC[2], (Senior Member, IEEE), PETER PEER[1], (Senior Member, IEEE), FRANC SOLINA[1], (Life Senior Member, IEEE), AND ALEŠ JAKLIČ[1], (Member, IEEE)

[1]Faculty of Computer and Information Science, University of Ljubljana, 1000 Ljubljana, Slovenia
[2]Faculty of Electrical Engineering, University of Ljubljana, 1000 Ljubljana, Slovenia

Corresponding author: Tim Oblak (tim.oblak@fri.uni-lj.si)

This research was supported in parts by the Slovenian Research Agency (ARRS) Project J2-9228 (B) "A neural network solution to segmentation and recovery of superquadric models from 3D image data" and ARRS Research Programs P2-0214 (B) "Computer Vision" and P2-0250 (B) "Metrology and Biometric Systems".

ABSTRACT Reconstruction of 3D space from visual data has always been a significant challenge in the field of computer vision. A popular approach to address this problem can be found in the form of bottom-up reconstruction techniques which try to model complex 3D scenes through a constellation of volumetric primitives. Such techniques are inspired by the current understanding of the human visual system and are, therefore, strongly related to the way humans process visual information, as suggested by recent visual neuroscience literature. While advances have been made in recent years in the area of 3D reconstruction, the problem remains challenging due to the many possible ways of representing 3D data, the ambiguity of determining the shape and general position in 3D space and the difficulty to train efficient models for the prediction of volumetric primitives. In this article, we address these challenges and present a novel solution for recovering volumetric primitives from depth images. Specifically, we focus on the recovery of superquadrics, a special type of parametric models able to describe a wide array of 3D shapes using only a few parameters. We present a new learning objective that relies on the superquadric (inside-outside) function and develop two learning strategies for training convolutional neural networks (CNN) capable of predicting superquadric parameters. The first uses *explicit supervision* and penalizes the difference between the predicted and reference superquadric parameters. The second strategy uses *implicit supervision* and penalizes differences between the input depth images and depth images rendered from the predicted parameters. CNN predictors for superquadric parameters are trained with both strategies and evaluated on a large dataset of synthetic and real-world depth images. Experimental results show that both strategies compare favourably to the existing state-of-the-art and result in high quality 3D reconstructions of the modelled scenes at a much shorter processing time.

INDEX TERMS Superquadrics, parametric models, reconstruction, 3D, deep learning, convolutional neural networks, CNN, parameter recovery.

I. INTRODUCTION

3D reconstruction represents one of the central problems of computer vision. It aims to interpret the shape, appearance, as well as the relative position of objects in the environment and to derive a unique (typically parameterized) description of the 3D world. In the context of artificial systems, a reconstructed scene can be used to inform an autonomous agent of its surroundings and to enable complex interactions, such as collision avoidance, maneuvering [1], [2] or grasping [3], [4]. While different approaches have been proposed in the literature for this task, bottom-up reconstruction of 3D scenes with volumetric primitives is particularly appealing. Such methods use a fixed vocabulary of possible elementary

The associate editor coordinating the review of this manuscript and approving it for publication was Carlos M. Travieso-González.

shapes to describe 3D scenes of arbitrary complexity, and therefore represent a highly flexible and descriptive approach to 3D reconstruction [5], [6].

The idea of a generalized bottom-up reconstruction originally came as a response to the advancements in perceptual psychology and neuroscience in the 1960's and 70's. The first theoretical vision system, introduced by Marr [7], heavily prioritized reconstruction and aimed at recovering 3D shapes from images by recognizing various depth cues and by fitting suitable volumetric models in a hierarchical manner. Biederman [5] argued, that human object recognition works by assembling specific volumetric primitives, called geons, into larger constellations, forming complex models of the environment. The transition from theoretical systems to working practical applications was driven largely by the choice of 3D representations. While highly-parameterized representations were able to ensure accurate 3D reconstruction, these came at the expense of computational overheads and complex scene descriptions. Using constellations of shape primitives (with comparably less parameters) to model 3D scenes and objects, on the other hand, resulted in more acceptable trade-offs between the reconstruction accuracy and the complexity of the scene description. Following this latter line of research, Barr first introduced so-called superquadrics [8] to the field of computer graphics and Pentland later brought them to the attention of the computer vision community [6]. Superquadrics are parametrized 3D models (or volumetric primitives), capable of forming a variety of different shapes with only a few shape parameters and represent a popular choice for 3D object representation. Considerable efforts have been directed towards recovering superquadrics from 3D data, e.g., [9]–[12], but the highly non-linear nature of the problem typically resulted in relatively slow optimization methods. Further development was ultimately burdened not only by this computational complexity, but also by the lack of affordable and efficient mechanisms to capture 3D data.

While the initial hypotheses from Marr [7] and Biedermann [5] lacked biological evidence, contemporary neuroscience literature generally acknowledges that complex shapes are represented as spatial arrangements of individual 3D parts in the human visual system [13]–[15]. The idea to construct more complex structures from a small set of basic elements is very powerful and is the foundation principle in many other scientific fields. As observed in the visual pathway of macaque monkeys, populations of neurons in specific regions of the cortex spike in response to certain 3D shapes, their positions and rotations [13], [15]. This neural activity was later successfully modeled with deep neural networks [16]. Our approach to the problem of 3D scene reconstruction using superquadrics as basic building blocks is thus heavily inspired by the current understanding of visual pathways in biological systems. This is also reflected in the choice of depth images as the input representation for our reconstruction model. As neurons seem to encode 3D spatial configurations of self-occluding 3D surface fragments, shape perception is not only based on 2D image processing [13]. It is now also well established that depth information is interpreted from binocular vision in the early stages of the visual cortex [17]. This information is then propagated towards the more complex areas of the cortex in a hierarchical feed-forward manner.

Recent works in the field of computer vision show signs of a revitalized interest in volumetric recovery with shape primitives and build mostly on advances in deep learning and specifically convolutional neural networks (CNNs) [18]–[21]. Novel solutions to this problem are typically constrained to labeled 3D datasets and lack the capacity to learn from partial-view data (such as depth images) only. The choice of learning objective also has a considerable effect on the final accuracy of the estimated model parameters and impacts the quality of the 3D reconstruction. A certain level of geometric awareness is needed to capture the properties of target objects. In this work we investigate the effect of various learning strategies and explore how different conceptual approaches to superquadric recovery affect the reconstruction process. Our aim is to revisit this problem with the use of CNNs to *(i)* develop recovery models that don't exhibit the computational overhead of early iterative solutions, such as [9], to *(ii)* ensure high prediction accuracy both in terms of superquadric parameter values as well as the scene reconstruction quality, and *(iii)* investigate different learning strategies for training deep learning models for superquadric recovery.

We focus on the problem of estimating the parameters of a single superquadric in a general position, which include parameters for size, shape, position and orientation of the superquadric. This estimation process represents a complex task due to the vagueness in describing spatial relations and especially, rotation. While a simple CNN regression model works for most superquadric parameters, as shown in [21], the real challenge lies in determining the rotation of the superquadric. Because symmetric superquadrics can be rotated arbitrarily along certain axes (e.g., consider a perfect sphere) without changing the appearance, rotation introduces an ambiguity into the recovery process that is difficult to address. Due to these symmetries we believe it is important to try to determine superquadric parameters using a geometry-aware learning criterion. To achieve this, we use the superquadric surface equation, which has several desirable characteristics and allows us to design a loss function that takes into account the shape and general position of the superquadrics. The optimization process with the considered loss is, hence, based on the 3D properties of superquadric surfaces, rather than the actual parameter values, which may be ambiguous. Gathering of 3D data can be an expensive and cumbersome process, especially when labeling individual examples manually. This requires real-world measurement, expert knowledge, or might even be impossible. It is therefore equally important to have at hand an approach that does not require a large labeled dataset, but is still able to learn a CNN predictor for superquadric recovery.

To address the challenges discussed above, we make the following contributions in this article:

- We introduce a novel geometry-aware learning objective based on the superquadric (inside-outside) function that allows us to train CNN predictors capable of predicting (with high accuracy) the parameters of a single superquadric in general position from input depth images. This learning objective improves on earlier work based on direct parameter regression, capable of only predicting parameters of unrotated superquadrics [21].
- Using the learning objective we propose two learning strategies to train the CNN predictor. The first uses *explicit supervision* and compares representations generated from the superquadric (inside-outside) function of the predicted and ground truth superquadric parameters. The second strategy relies on *implicit supervision* and instead utilizes a differentiable renderer to construct a depth image from the predicted parameters, which is compared directly to the input depth image. No ground truth parameters are needed during learning for this second strategy.
- We evaluate our learning objective, and both learning strategies in rigorous experiments on a dataset of more than 150.000 depth images containing various superquadrics. We present an extensive analysis of the generated results and compare our solutions to the state-of-the-art in superquadric parameter recovery.

II. RELATED WORK

This section provides a brief overview of existing superquadric reconstruction techniques and discusses relevant techniques used in 3D-oriented deep learning.

A. SUPERQUADRIC RECOVERY

Superquadric recovery refers to the problem of estimating the parameters of the superquadric volumetric primitives, such that the primitives describe the input data as well as possible.

1) EARLY METHODS

Pentland first proposed a brute-force search of the superquadric parameter space [22] using parallel computing. He tried to recover superquadrics from color images by analysing shading information, but had limited success. In 1987, Solina and Bajcsy [23] devised a least-squares minimization process for superquadric recovery from range images, which they further refined a few years later [9]. To fit the models, they used the superquadric inside-outside function, which explicitly describes the relationship between a point and the superquadric surface. An alternative solution to this problem was proposed by Boult and Gross in [10], [24]. Here, the authors proposed using the superquadric radial distance, which approximates the distance from a given point to the superquadric surface, as the fitting function. However, the recovered superquadrics were visually almost identical to those, recovered by the inside-outside function [25]. Because calculation of the radial distance was computationally more demanding than using the inside-outside function, the latter was used more extensively by researchers working in this area [9], [26]–[28]. Certain superquadric shapes, particularly those with sharp edges, can cause the inside-outside function to become susceptible to numerical overflow and singularities. To alleviate this, Vaskevicius and Birk [29] introduced a numerically stable method of computing the gradient of the inside-outside function with respect to the superquadric parameters, making possible to better model shapes, such as cuboids and cylinders. Others have approached the recovery procedure from another perspective, for example, by using genetic algorithms [11]. Extensions to superquadrics were also proposed [30], [31], but ultimately these also relied on iterative optimization procedures during recovery, which stalled further development in this direction.

The original method by Solina and Bajcsy [23], designed for the recovery of isolated superquadrics, was later extended by Leonardis *et al.* [12] to handle more complex shapes that needed to be modelled by multiple superquadrics. The authors achieved joint segmentation and recovery of multiple superquadrics on the basis of the Minimum Description Length (MDL) principle, by using a bottom-up approach of splitting the shape into multiple parts and then fitting superquadrics to shape parts individually. The fitting function, however, remained of iterative nature. Chevalier *et al.* [32] tackled the same problem using a top-down approach, by splitting the shape iteratively until the subshape was viable for superquadrics representation.

2) DEEP LEARNING METHODS

More recent work on the recovery of volumetric primitives is mostly based on deep learning techniques. In 2017, for example, Tulsiani *et al.* [18] proposed a method to learn shape abstractions using primitive shapes. The authors used cuboids as their geometric primitive of choice and fitted them to triangle meshes. In a recent article by Paschalidou *et al.* [19], the authors adapted this pipeline to use superquadrics instead of cuboids, achieving a significantly smaller fitting error due to the wide range of shapes that superquadrics can approximate. A later expansion to this work [20] proposed a system for hierarchical unsupervised recovery of superquadrics. Both of the above works used labeled 3D data with specific object categories, e.g., planes, animals, chairs, etc., to train CNN superquadric predictors. However, to make recovery techniques applicable to arbitrary data a more generalized approach is needed that does not necessarily rely on explicit supervision. In [21], Oblak *et al.* presented a CNN predictor for (unrotated) superquadric recovery, capable of estimating superquadric parameters from a depth image in a single forward pass of the model without the need for costly iterative optimization techniques used in the early work in this field [9]. Oblak's work was later extended by Šircelj *et al.* to include the capability of recovering multiple super-quadrics [33]. Superquardics were also recovered from point clouds by Slabanja *et al.* [34], but all techniques again relied on ground truth parameters during training.

The usefulness of superquadrics was already proven in practical applications, especially in various robot grasping tasks [4], where the shape and position of objects is undeterministic, or, for example, for handling of mail pieces [23]. Another recent example of their use is in heritage science for 3D documentation of artifacts [35], [36]. For a comprehensive coverage of the field, the reader is referred to [28] for an excellent survey on this topic.

B. DEEP LEARNING IN 3D

Contemporary techniques for superquadric recovery are closely related to recent techniques for 3D-based deep learning. Below we discuss topics from this field that are also relevant for the problem addressed in this work, that is, the choice of 3D representation and the issue of pose estimation.

1) 3D REPRESENTATION

The choice of 3D representation is an important factor to consider for various types of input, intermediate and output data. Depending on the task at hand, a combination of these representations is used in the literature [37]–[41]. Wu *et al.* [37] were the first to introduce the idea of using discretized volumetric grids for spatial representation and utilize 3D encoders to process volumetric data. In another work, Wu *et al.* proposed MarrNet [38], an end-to-end trainable framework, which takes as input only a single RGB image, as an intermediary step estimates surface normals, depth and the silhouette of the object and finally predicts the 3D shape of the object using a 3D decoder. Volumetric grids can describe various spatial data, such as signed distance functions [40] or occupancy functions [39], [41]. It is obvious, that 3D encoders have a far greater impact on memory consumption and performance than 2D encoders. Nevertheless, volumetric grids allow for storage of true 3D data, whereas 2D images only contain a single perspective, leading to object self-occlusion and loss of information. Different from the works discussed above, our model relies on depth images, a type of 2.5D representation, which encodes 3D information into a 2D structure. This allows us to use 2D encoders to process the data efficiently in contrast to 3D encoders, while greatly benefiting from the more explicit spatial description in comparison to a color image. The mechanisms to capture depth images have also become more affordable in recent years and are being incorporated into various consumer products with an ever smaller form factor, e.g., smartphones.

2) POSE ESTIMATION

Recent pose estimation models rely on CNN-based predictors that generated pose estimates in the form of continuous parameter values [42]–[44]. Zhu *et al.* [43], for example, use a standard encoder-decoder architecture to reconstruct volumetric data and simultaneously train a pose regressor. Miao *et al.* [42] use a Mean Squared Error (MSE) based loss to train 6 separate regressors for all parameters describing object pose, i.e., position and rotation parameters. While these techniques produce solid pose estimates, methods based on loss function that include a geometric component typically achieve stronger results. For example, Xiang *et al.* [44] minimize the distance between points on the surface of rotated objects to predict rotation and demonstrate impressive performance. We strongly believe this geometric awareness is crucial when learning representations in the spatial domain. Additionally, the choice of rotation description also plays a major role in the success of pose estimation techniques. For example, Euler angles are known to suffer from gimbal lock, whereas rotation quaternions have the unit norm constraint, which makes regression a non-trivial task [43]. The problem of superquadric recovery addressed in this article is related to the methods discussed above in that it also requires pose estimation and prediction of translation and rotation parameters. In line with the most successful techniques in this domain, our solutions also use a geometry-aware loss during training, but are designed specifically for the problem of superquadric recovery.

III. LEARNING CNN PREDICTORS FOR SUPERQUADRIC RECOVERY

A critical component of bottom-up 3D scene reconstruction techniques is an efficient estimation of the volumetric primitives. In this section we present two CNN-based solutions to this problem that are able to recover superquadric models from depth images. As illustrated in Fig. 1, the first solution uses a learning approach that relies on explicit supervision defined over the superquadric function, whereas the second exploits an implicit approach utilizing a differentiable renderer to learn a predictor of the model parameters. In the following section, we first present a formal description of superquadrics and then describes both solutions proposed in this work.

A. THEORETICAL BACKGROUND AND PROBLEM FORMULATION

Superquadrics represent volumetric primitives defined by a so-called inside-outside function. The function is defined for each point $p_s = [x, y, z]^T$ in object-space given by the following equation:

$$F(x, y, z) = \left(\left(\frac{x}{a_1} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{a_2} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_3} \right)^{\frac{2}{\epsilon_1}}, \quad (1)$$

where a_1, a_2, and a_3 determine the size of the superquadric in each axis of the coordinate system, and the parameters ϵ_1 and ϵ_2 define the shape of the superquadric. A set of superquadrics with various values of ϵ_1 and ϵ_2 and fixed size parameters can be observed in Fig. 2.

The inside-outside function is defined in local (i.e., superquadric centered) coordinates, but a more convenient way is to evaluate the function in world-space coordinates $p_w = [x_w, y_w, z_w]^T$. Thus, a transformation is needed to transform world coordinates p_w into local coordinates $p_s = [x_s, y_s, z_s]^T$. To do this, the inverse of the homogeneous

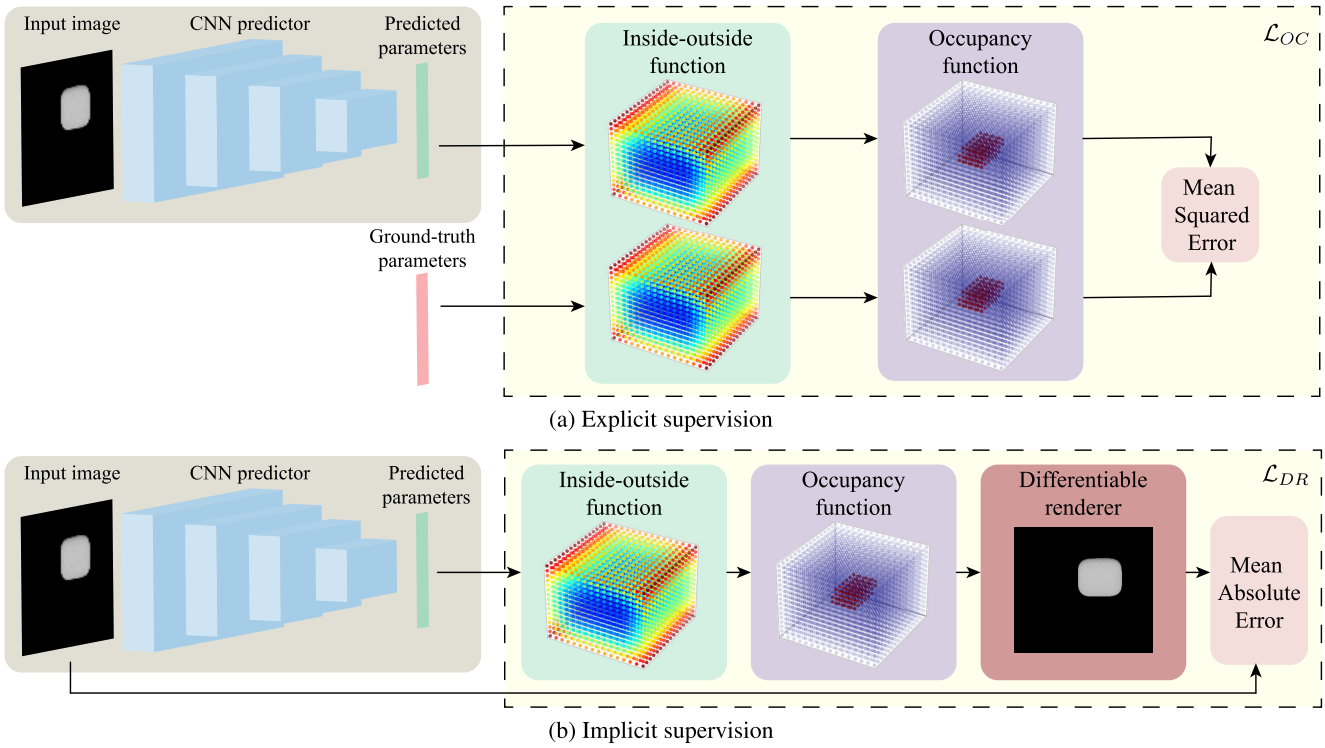


FIGURE 1. **Proposed learning strategies and loss functions. We propose two strategies to train CNN predictors for superquadric recovery in this article. Both strategies are based on the superquadric inside-outside function: (a) *Explicit supervision*: this learning strategy uses the ground-truth and predicted superquadric parameters to compute 3D occupancy grids over which a learning objective is defined. (b) *Implicit supervision*: this learning strategy requires no ground truth superquadric parameters and instead uses a differentiable renderer to reconstruct a depth image for the predicted superquadric parameters; the rendered and input depth images are then compared to provide a supervisory signal for the learning procedure. The figure is best viewed in color.**

transformation matrix M can be used, i.e.,

$$M^{-1} = \begin{bmatrix} r_{11} & r_{21} & r_{31} & -(t_1 r_{11} + t_2 r_{21} + t_3 r_{31}) \\ r_{12} & r_{22} & r_{32} & -(t_1 r_{12} + t_2 r_{22} + t_3 r_{32}) \\ r_{12} & r_{23} & r_{33} & -(t_1 r_{13} + t_2 r_{23} + t_3 r_{33}) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where $r_{i,j}$, for $i, j \in \{1, 2, 3\}$, are elements of a 3×3 rotation matrix and $t = [t_1, t_2, t_3]^T$ is a translation vector. Local coordinates can then be computed using

$$p_s = M^{-1} p_w, \tag{3}$$

which allows for the evaluation of the inside-outside function in world-space coordinates, i.e., $F(M^{-1}p_w)$. Note that homogeneous coordinates are utilized to calculate the transformation and that the inside-outside function only receives a local 3D point as input. To streamline the notation we use $F(p)$ hereafter to denote the inside-outside function, but note that world-centered points p_w are used always as input. As a result of this formulation, 12 parameters are needed to uniquely define a superquadric $F(p; \lambda)$:

$$\lambda = (a_1, a_2, a_3, \epsilon_1, \epsilon_2, t_1, t_2, t_3, q_1, q_2, q_3, q_4), \tag{4}$$

where q_1, q_2, q_3, and q_4 are rotational parameters, which correspond to the coefficients of a unit quaternion.

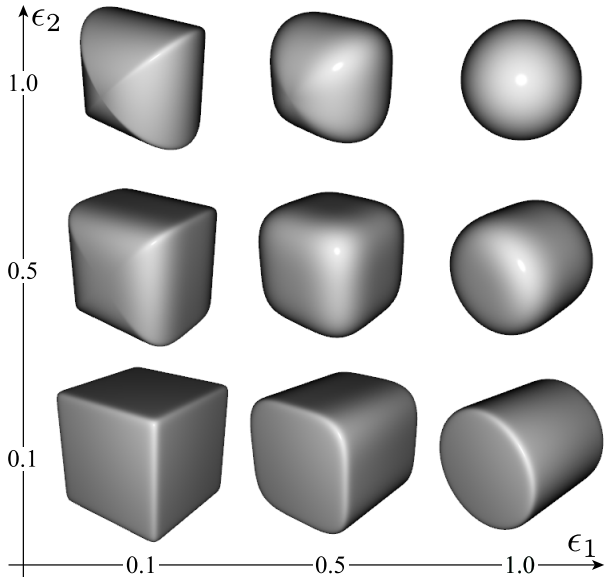


FIGURE 2. **The superquadric vocabulary. By changing the value of the parameters $\epsilon_{1,2}$, superquadrics can form many different shapes, from cuboids to ellipses and everything in between.**

The quaternion can be trivially transformed into a rotation matrix by using the Euler-Rodriguez formula [45].

For a point $p \in \mathbb{R}^3$, it is possible to determine it's position in relation to the superquadric surface. If the point lies inside

the superquadric, then $F(p) < 1$, if it lies outside of the superquadric, then $F(p) > 1$ and if $F(p) = 1$, then the point lies on the surface of the superquadric model. The value of F at the center of the superquadric is 0. The inside-outside function $F : \mathbb{R}^3 \rightarrow \mathbb{R}^+$ is continuous and differentiable, though it is numerically unstable for low values of the shape parameters $\epsilon_{1,2}$. This is usually alleviated by constraining the parameters to $\epsilon_{1,2} > 0.1$ [9]. A recent work by Vaskevicius and Birk [29] introduced a numerically stable formulation of the gradients, even for 0 value shape parameters.

To estimate the parameters λ of a superquadric model from an input depth image X, we present in this work two CNN learning strategies, such that the learned models f_{CNN} produce parameter estimates as close to the reference values as possible, i.e.:

$$\hat{\lambda} = f_{CNN}(X; \theta), \tag{5}$$

where $\hat{\lambda}$ is the CNN output and θ are learnable parameters of the network. Since superquadric parameters λ are continuous real values, we formulate this this task as a problem akin to regression.

B. EXPLICIT SUPERVISION

The first approach to superquadric recovery proposed in this article uses explicit supervision to estimate the parameters λ, as illustrated in the top row of Fig. 1. The assumption here is that ground truth parameters λ are available for every training image X and the goal is to learn a CNN predictor that produces parameter estimates $\hat{\lambda}$ that are as close as possible to the available ground truth. A direct minimization over the estimated and ground truth parameters does not result in representative parameter errors and consequently fails to capture the ambiguous nature of rotation in 3D space. Thus, the first approach minimizes the difference between 3D occupancy grids of the predicted and reference superquadrics.

If evaluated for every point in space, the result of the inside-outside function is a superquadric hypersurface in $\mathbb{R}^4$, which can be used as an indicator of the model error during training. The biggest differences between the predicted and reference hypersurfaces occur outside of the superquadric where $F(x, y, z) \gg 1$. To focus on the differences in close proximity to the superquadric, which impact the model learning procedure the most, the inside-outside function is further transformed into a differentiable occupancy function. We follow the proposal of [20]:

$$G(x, y, z) = \sigma(s(1 - F^{\epsilon_1}(x, y, z))), \tag{6}$$

where $G : \mathbb{R}^3 \rightarrow (0, 1)$ and s is a scaling factor, which controls the sharpness at the spatial border of the superquadric. This function returns a value close to 1 if a point is inside the superquadric, close to 0 if it is outside and 0.5 if the point is directly on the surface of the superquadric. The function is continuous and therefore differentiable. Note also that the inside-outside function is raised to the power of ϵ_1 before computing the occupancy function, as suggested in [28]. This operation does not change the surface itself, but ensures that all model parameters contribute to a similar extent to the overall prediction error and the learning procedure is not dominated by the shape parameters.

An approximation of this hypersurface can be computed by first discretizing the coordinate system into a set of fixed, equally-distanced points. The discretization procedure is controlled by a resolution parameter r and the minimum and maximum bounds for each of the axes, b_{min} and b_{max}, respectively. We sample r equally spaced points in each axis from b_{min} to b_{max}, which results in a 3D grid of discretized points. For each of these points, the occupancy function is evaluated and stored in a volumetric grid:

$$V_{G,\lambda}(i, j, k) = G(x_i, y_j, z_k; \lambda); \quad i, j, k = 1, 2, \ldots, r, \tag{7}$$

where x_i, y_j, z_k denote the coordinates of the discretized points in world space. By doing this, a discretization $V_{G,\lambda}$ of the 3D occupancy function in Eq. (6) is obtained for the given parameters λ. In other words, a voxel grid is created, where each voxel encodes the value of the occupancy function at that location. The size of the grid corresponds to the selected resolution: $V_{G,\lambda} \in \mathbb{R}^{r^3}$.

To learn the parameters θ of the CNN predictor f_{CNN} the first approach studied in this article uses the Mean Squared Error (MSE) to calculate the difference of two differentiable occupancy grids:

$$\mathcal{L}_{OC}\left(\lambda, \hat{\lambda}\right) = \frac{1}{|V|} \sum_{i,j,k}^{r} (V_{G,\lambda}(i, j, k) - V_{G,\hat{\lambda}}(i, j, k))^2, \tag{8}$$

where λ are ground truth parameters of the target superquadric and $\hat{\lambda}$ are estimated superquadric parameters. In other terms, the sum of all the squared differences between matching points is computed first and then divided by the size of the grid $|V|$.

C. IMPLICIT SUPERVISION

The second proposed approach to superquadric recovery uses implicit supervision to predict superquadric parameters. As illustrated in Fig. 1 (b), no explicit reference value for the superquadric parameters is provided with this approach. Instead a depth image is used as input to the prediction procedure and a loss function is defined for the learning stage between the input depth image and the depth image rendered based on the current estimate of the superquadric parameters. Such an approach eliminates the need for ground-truth parameters and is able to learn a CNN predictor directly by comparing the input and rendered shapes.

To derive a *self-supervised* loss function for model learning, a differentiable way of reconstructing depth images from the predicted superquadric parameters is needed. While different techniques have been presented in the literature for this purpose, the depth projection operator of Gadelha *et al.* [46] assumes an orthographic projection, which makes it particularly suitable for our approach, since the depth images used in this work are rendered orthographically. The algorithm takes a volumetric voxel grid as input and (from a specific view)

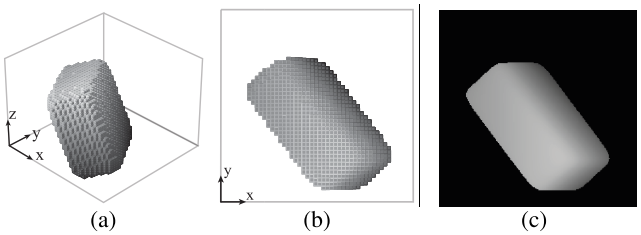


FIGURE 3. The depth projection operator. Following the procedure of Gadelha *et al.* [46], this work relies on a differentiable renderer to reconstruct depth images of superquadrics in orthographic projection from predicted parameters. Visualized are the key 3D representations in the process: (a) A superquadric, represented by a volumetric voxel grid, rendered in an isometric projection, (b) the same voxel grid, viewed perpendicular to the *xy*-plane and (c) a depth image of the superquadric. The differentiable renderer takes a voxel grid (a) as input, then projects the depth onto a specific plane (b), which results in a depth image (c), similar in appearance to the depth images in the experimental dataset. Note that a 64^3 space is used to visualize the voxel grid, while the depth image shown is rendered in a space of size 256^3. During training, these resolutions are matched.

projects the depth for each line of sight onto a 2D surface. Analogous to a voxel grid, the continuous superquadric occupancy grid from Eq. (7) is taken and projected onto a 2D plane perpendicular to the z axis. This procedure is illustrated in Fig. 3.

To implement the procedure, we follow the formulation of Gadelha *et al.* [46], however, some adjustments are also made to the process, so the output matches the input depth images as closely as possible. The value of an element in the occupancy grid is either close to 0 at indices outside of the superquadric or close to 1 at indices inside the superquadric. The sharpness of transition close to superquadric surface is determined by the parameter s in Eq. (6). To calculate the depth projection, the authors define an intermediate function $A : \mathbb{R}^{n^3} \rightarrow \mathbb{R}^{n^3}$:

$$A(V, i, j, k) = \exp(-\tau \sum_{l=1}^{k} V(i, j, l)), \tag{9}$$

where $V : \mathbb{Z}^3 \rightarrow (0, 1)$ is the occupancy grid, (i, j, k) are grid indices and τ is a parameter, which controls the sharpness of transition between empty space and the object. With the term $\sum_{l=1}^{k} V(i, j, l)$, the cumulative sum of voxels along each line of sight for each index k is calculated. The cumulative sum is (for each index) then passed to an exponential function, which generates A values of 1 until a voxel on the surface of the superquadric is hit, since the sum is 0. When a surface voxel is hit, the cumulative sum becomes greater than zero, driving the exponential function close to zero. Again, the sharpness of this transition is determined by parameter τ.

To render a depth image, we sum all values along the z axis for each line of sight. Each of these values then represents the distance from the near plane of the view frustum to the first intersected surface in the line of sight. The original formulation from [46] implies that the background has infinite distance from the near plane of the viewing frustum. In our case, the space is bounded into a r^3 grid, so appropriate modifications are made to the method. Specifically, to calculate depth the following operator $T : \mathbb{R}^{r^3} \rightarrow \mathbb{R}^{r^2}$ is utilized:

$$T(V) = 1 - \frac{1}{r} \sum_{k} A(V, i, j, k), \tag{10}$$

where r is the resolution of the voxel grid in a single axis. The accumulated depth is divided by the resolution of the voxel grid to normalize the depth values to the range [0, 1]. A value of 1 then represents a distant point and a value of 0 represents a point close to the observer. To match this with the depth images in the dataset used for our experiments, all generated values are subtracted from 1. For a given set of superquadric parameters, the depth image rendered with the presented approach is a close approximation of the depth images present in the experimental dataset.

The overall loss function for this approach is then defined as the Mean Absolute Error (MAE) between the input depth image and the rendered depth image, constructed from the predicted superquadric parameters:

$$\mathcal{L}_{DR}\left(X, \hat{y}\right) = \frac{1}{|X|} \sum_{i,j}^{r} |X(i, j) - T(V_{G,\hat{y}})(i, j)|, \tag{11}$$

where $V_{G,\hat{y}}$ is the occupancy grid, r is the size of the image in one axis and (i, j) are pixel indices of the image. Note, that the input depth image X should be appropriately resized to match the resolution r^2 of the reconstructed depth image. An L_1-based error measure is used, to reduce the impact of outliers on the loss function.

D. THE CNN PREDICTOR

Both learning approaches described in the sections above use a CNN model to predict superquadric parameters from input depth images. Several models can be used here, but a modified ResNet [47] is selected for this work. The modular nature of the model allows adapting the model depth in accordance with the complexity of the prediction task.

Because in our case the input of the model receives rather simple depth images with isolated superquadrics, we select a shallow ResNet-18 [47] as the CNN predictor, as shown in Fig. 4. The first convolutional layer of the model has a filter of size 7 to capture a bigger initial receptive field. This is needed, as the input depth images contain mostly low-frequency information. From that point on, a filter of size 3 is used with all subsequent layers. By setting the stride of convolution to 2 every 3 convolutional layers, the data is pooled, which widens the receptive field of the convolutional filters. At the top of the network a prediction head with two 256-dimensional fully-connected layers is added. This head mixes and processes features, received from the convolutional layers. The network output is then split into four groups, one for each parameter type. Size, shape and translation parameter groups each have a fully-connected layer with 3, 2 and 3 outputs, respectively. To these, a final sigmoid activation is attached, which is done so the initial predictions result in values close to "0.5", making the result

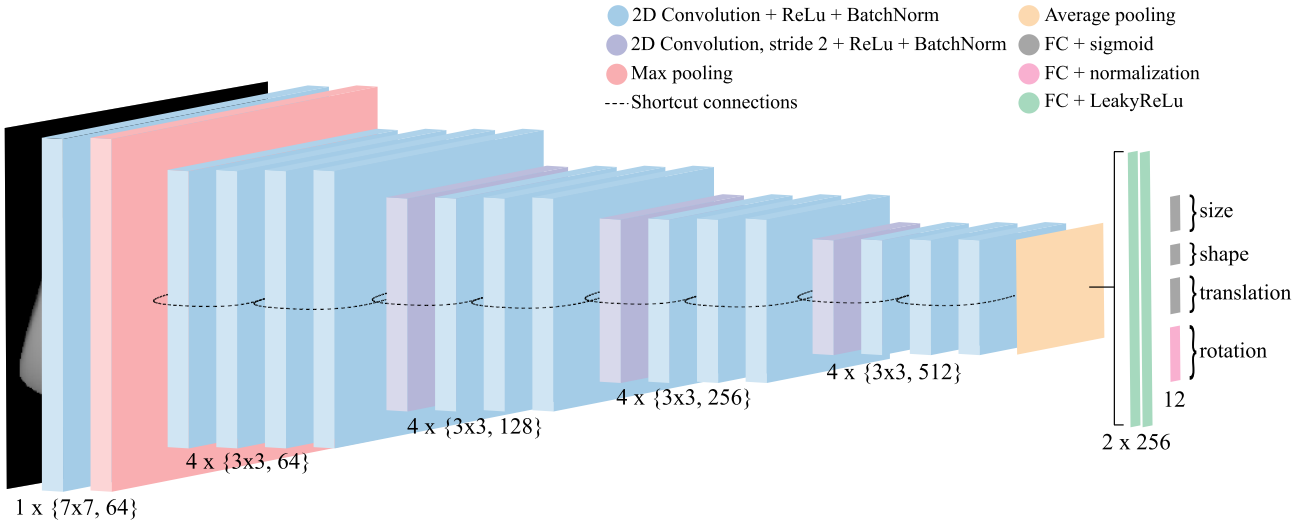


FIGURE 4. The CNN predictor. The figure illustrates the architecture of the CNN model used to predict superquadric parameters from input depth images. We use the $X \times Y, Z$ notation to denote a convolutional layer with Z filters of size $X \times Y$. A modified Resnet-18 model is used, which outputs a 12-dimensional vector of superquadric parameters, defining the general position, scale and appearance of the 3D shape in the input range image. The figure is best viewed in color.

of the superquadric inside-outside function stable and suitable for backpropagation. The predicted values for size and translation are multiplied by 256 to scale them back into the 256^3 space. Rotation parameters have a final fully-connected layer with 4 outputs. Since versors (i.e., unit quaternions) are used to describe rotation, L_2-based normalization is selected as the final activation function. To speed up the training process, transfer learning is utilized. Specifically, pre-trained weights for ResNet-18, trained on the ImageNet [48] dataset, are loaded during model initialization.

IV. EXPERIMENTS

This section presents the experiments conducted to validate the proposed learning strategies. The section starts with a description of the experimental setup, dataset, performance metrics and the training procedure and then proceeds to the actual experiments and discussion of the main results and findings.

A. DATASET

For the experiments we use a synthetic dataset, similar to [21]. The dataset contains artificially generated depth images, where a single superquadric is placed in a scene and rendered in orthographic projection using a custom ray-tracing renderer. The renderer works by following a ray in discrete steps along the z axis and by detecting the intersection with the superquadric surface. When the intersection is detected, the distance from the viewport to the superquadric surface is used as the depth value of the corresponding pixel. The result of this rendering procedure is an image of size 256×256. This is convenient, since raw grayscale pixels are usually stored with 8 bits of information, which yields a total of 256 possible pixel values. This way, the depth image represents a 3D space of size 256^3.

The training part of the dataset consists of 150.000 depth images. Each image is annotated with 12 parameters that encode the shape, size, translation and rotation of the rendered superquadrics. For the test set, an additional set of 20.000 examples is generated and used for the final performance evaluation and comparison with competing models from the literature. To generate the dataset, the values of the superquadric parameters are sampled from $\mathcal{U}(25, 75)$ for the size parameters, from $\mathcal{U}(88, 168)$ for the translation parameters and from $\mathcal{U}(0.1, 1)$ for the shape parameters. The Subgroup algorithm [49] is used to draw a random uniform rotation in the form of a quaternion. A selection of examples can be seen in Fig. 5.

B. METRICS

We evaluate the two CNN predictors by comparing the predicted parameters to the ground truth parameters using the Mean Absolute Error (MAE). The order of the predicted size and shape parameters can be arbitrary at the output of the CNNs due to the ambiguity of the superquadric shape description [9]. For example, an unrotated cuboid with size parameters $a_{1,2} = 1, a_3 = 2$ is visually identical to a cuboid with parameters $a_{1,3} = 1, a_2 = 2$, which is rotated by 90° around the x axis in the local coordinate system. A similar property can be observed with the shape parameters $\epsilon_{1,2}$. Due to this uncertainty, we average over the elements of the size and shape parameter groups, giving us only one value for the size parameter subset $\overline{a}$ and $\overline{\epsilon}$ for the shape parameter subset.

$$\overline{a} = \frac{1}{3}\sum_{i}^{3} a_i \qquad \overline{\epsilon} = \frac{1}{3}\sum_{i}^{2} \epsilon_i \tag{12}$$

We omit rotational parameters in this evaluation as rotation can be particularly ambiguous, i.e., different quaternions can result in the same superquadric surface. For example,

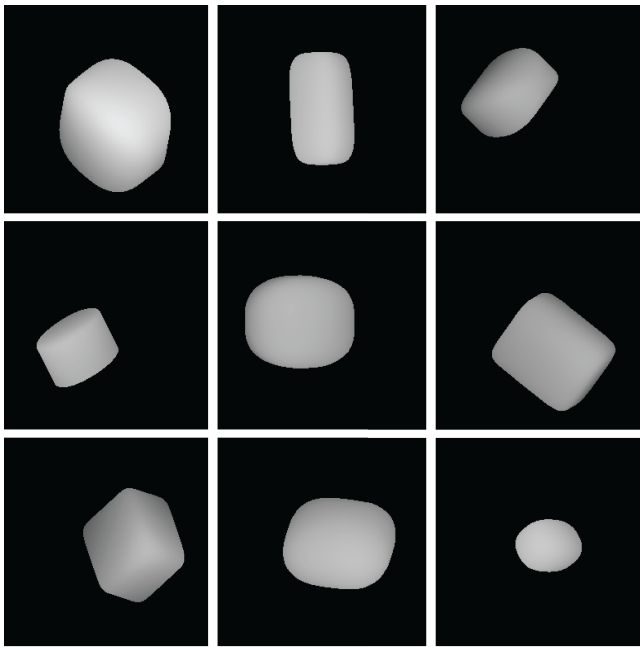


FIGURE 5. Examples in our superquadric dataset. Depth images contain a single superquadric inside a 256^3 space, rendered in a orthographic projection. Higher pixel values represent surfaces closer to the viewer and black pixels represent the background.

an arbitrary rotation can be applied to a perfectly spherical superquadric with parameters $a_{1,2,3} = 1, \epsilon_{1,2} = 1$ and the resulting shape would be visually identical to the original shape.

To have a measure that compares superquadrics in a geometric manner, the Intersection over Union (IoU) is reported in the experiments. Specifically, a special variant of IoU, defined inside 3D space, is adopted. Using the binary occupancy function

$$B(x, y, z) = \begin{cases} 1; & F^{\epsilon_1}(x, y, z) \leq 1 \\ 0; & F^{\epsilon_1}(x, y, z) > 1 \end{cases} \tag{13}$$

a superquadric voxel grid $V_{B,\lambda}$ is generated. All points that lie outside of the superquadric then have a value of 0 and all outside have a value of 1. The IoU metric is then calculated between binarized voxel grids of predicted and ground-truth superquadric parameters, calculating the number of voxels that are surrounded by both superquadrics divided by the number of voxels surrounded by either one of them, i.e.:

$$IoU(\hat{\lambda}, \lambda) = \frac{\sum_{i,j,k}^{r} V_{B,\hat{\lambda}}(i,j,k) \wedge V_{B,\lambda}(i,j,k)}{\sum_{i,j,k}^{r} V_{B,\hat{\lambda}}(i,j,k) \vee V_{B,\lambda}(i,j,k)}, \tag{14}$$

where λ are ground-truth parameters and $\hat{\lambda}$ are the predicted parameters. The goal of superquadric recovery is to maximise this performance measure, which represents the ratio of coverage between the generated and true superquadrics. A value of 0 means that there is no overlap and a value of 1 means that the predicted and ground truth superquadrics overlap perfectly.

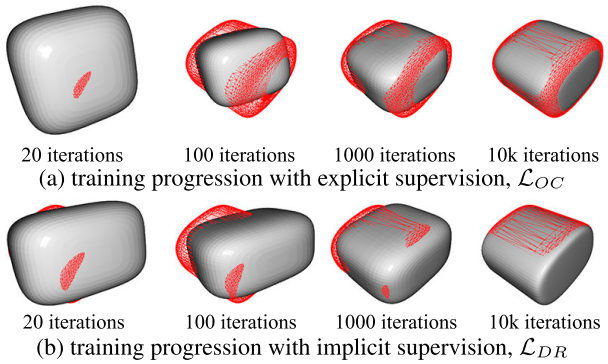


FIGURE 6. Parameter fitting during training. We visualize on a sample superquadric the fitting procedure during training for both learning strategies, that is, with explicit and implicit supervision. Size and translation parameters are learned first, followed by rotation and finally shape.

The IoU measure also enables us to indirectly evaluate the prediction capability for the rotation, which is not possible with the simple MAE metric used for the rest of the parameters. Since IoU is dependent on the overlap between the predicted and ground truth surface, it is robust to orthogonal or 180° rotations along the superquadric axes.

C. TRAINING PROCEDURE

The experimental dataset described in Section IV-A is split into three parts: 135.000 images that are used for training the CNN predictors, 15.000 images for validation and another 20.000 images to perform the final tests and report results. During each epoch, we iterate through the whole training dataset and then evaluate the performance on the validation set. The data is also shuffled in each pass to ensure a representative distribution of the whole dataset in a single batch. For the optimization algorithm, the Adam [50] optimizer is used with an initial learning rate of 1e−4 and batch size of 32 in accordance with standard methodology [51]–[53]. To stabilize the gradient descent around the local minimum and to ensure convergence of the loss function, the learning rate is decreased by a factor of 10, when the validation loss stagnates for 10 epochs. The training procedure is stopped when the validation loss fails to improve for more than 20 epochs.

To calibrate the renderer (Section III-C) needed for the learning strategy with implicit supervision, the two sharpness hyper-parameters, s (from Eq. (6)) and τ (from Eq. (9)) are set by conducting a grid search over the parameter space (using the training data). Ultimately, the parameters are set to $\tau = 4.8$ and $s = 117$, so that the rendered images differ as little as possible from the ones in the dataset for the same set of superquadric parameters. There is also the resolution parameter r, which determines the granularity of any 3D representation used in our learning criterions, e.g., voxel grids and depth images. Higher resolutions lead to better approximation of a continuous 3D space and smoother loss functions, but lower resolutions result in faster training times. A value of $r = 32$ is used for the explicit supervision method and $r = 64$ for the implicit supervision.

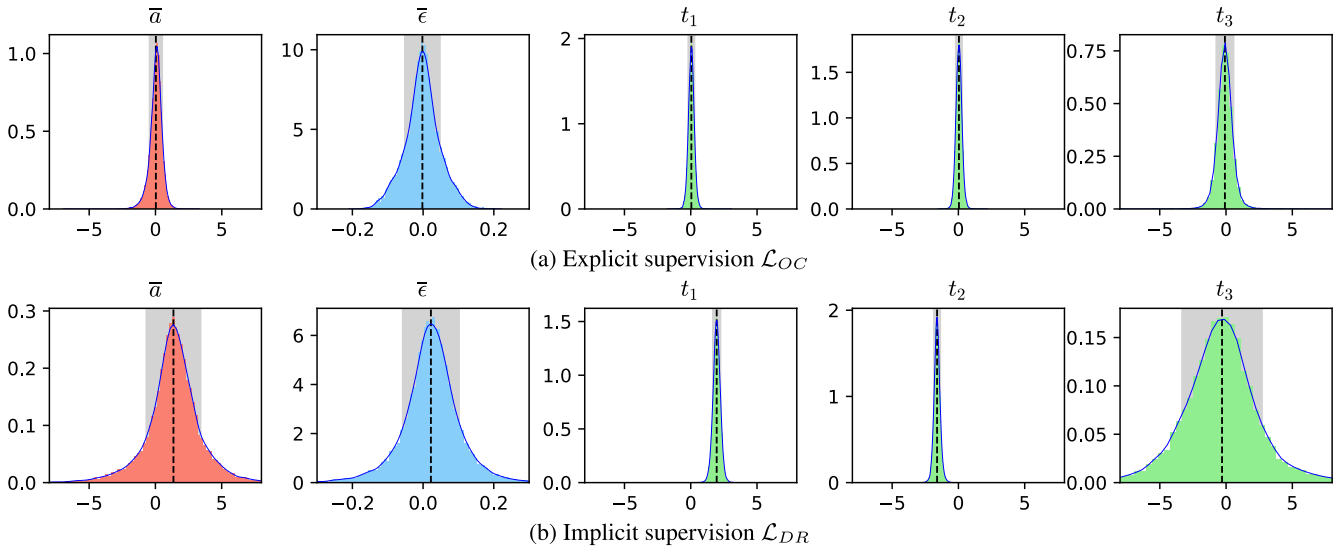


FIGURE 7. Error distributions for the predicted parameters. Both learning strategies result in CNN predictors that are capable of predicting superquadrics with a reasonable level of accuracy. The models learned with explicit supervision, however, produce somewhat more accurate and more consistent estimates with the considered experimental dataset. The figure is best viewed in color.

TABLE 1. Parameter prediction accuracy of the learned models. The table reports Mean Absolute Errors (MAE) and IoU accuracy of the predicted parameters computed over all test images. Note that the errors of the shape (ϵ) and size (a) parameters are averaged over the individual parameters (i.e., over a_1, a_2 and a_3 for shape and over ϵ_1 and ϵ_2 for size), since their predicted order is arbitrary.

Method	Size [0-256]	Shape [0-1]	Translation [0-256]			IoU [%]
	$\overline{a}$	$\overline{\epsilon}$	t_1	t_2	t_3	
Explicit supervision $\mathcal{L}_{OC}$	$\mathbf{0.35 \pm 0.31}$	$\mathbf{0.03 \pm 0.03}$	$\mathbf{0.17 \pm 0.14}$	$\mathbf{0.18 \pm 0.14}$	$\mathbf{0.45 \pm 0.47}$	$\mathbf{94.62 \pm 3.18}$
Implicit supervision $\mathcal{L}_{DR}$	1.94 ± 1.49	0.06 ± 0.05	1.95 ± 0.28	1.61 ± 0.23	2.19 ± 2.08	85.64 ± 5.72

During training, the CNN predictor learns some parameters faster than others, as visualized in Fig. 6 for a sample image from the training part of the dataset. The model trained with explicit supervision first learns the position and size of the superquadric. This is achieved in about 100 iterations. Then, the superquadric is slowly rotated into the appropriate general position. Shape parameters are fitted last, at a slow pace, requiring the model to complete more than $10k$ iterations to converge to a local minimum. This behaviour changes for the model trained with implicit supervision. With this approach, parameters are fitted first to match the 2D contour of the reconstructed depth image to the contour of the input depth image. As a result of this initial fitting, pixels that contain the superquadric in one depth image, but are part of the background in the other, now have the biggest impact on the learning objective. This effect is consequently minimized next. The size and rotation of the superquadric then gradually converge in about 1000 iterations. As with the explicitly learned model, the shape of the superquadric is learned last. The choice of 3D representation used in loss functions determines the shape of parameter space during training. When using depth images, the optimization is initially guided by differences in 2D features, such as object contour. In contrast, parameters are optimized more equally when comparing 3D data representations.

D. RESULTS AND DISCUSSION

This section presents quantitative and qualitative results obtained with the CNN predictors learned with the two proposed learning strategies. Comparisons with state-of-the-art methods from the literature are also reported.

1) MODEL EVALUATION AND ANALYSIS

The first series of experiments aims at evaluating the performance of the proposed CNN predictors on the test data of the experimental dataset.

Table 1 shows the comparison of the two proposed models in terms of prediction accuracy and Fig. 7 shows the error distribution over the predicted parameters. As can be seen, the model learned with explicit supervision performs best and exhibits stable behaviour with Gaussian-like error distributions for all parameter groups. No major biases are induced into the model, since most parameters are centered around an error of 0. Most parameters also exhibit limited prediction-error variance, which suggest the parameters are valid and close to the ground truth for most of the images in the dataset. The behaviour of the model trained with implicit supervision is more irregular. The predicted superquadrics have on average a slightly bigger volume, which is indicated by the positive average error of the size parameters $\overline{a}$ shown in Fig. 7. The shape of the predicted superquadrics

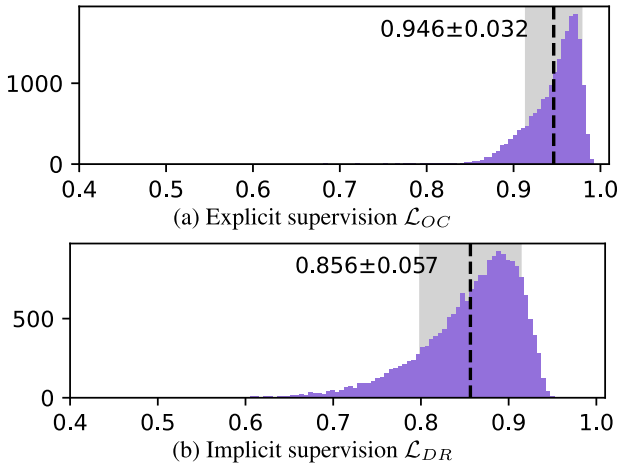


FIGURE 8. Distribution of IoU values. The graphs show a comparison of the IoU scores generated on the test images by the CNN predictors trained with (a) explicit and (b) implicit supervision.

is also more rounded in comparison to the ground-truth - positive average error of $\overline{e}$ in Fig. 7. The center coordinates t_1 and t_2 exhibit a bias in the positive and negative direction, respectively. This suggests that the predicted superquadrics are positioned somewhat towards the upper right corner of the reconstructed depth image, though the displacement is relatively small in relation to the whole 3D space. If we compare the distribution of the third coordinate t_3 between the models learned with explicit and implicit supervision, we observe a large disparity in the error variance. The implicit loss function has to derive depth information from pixel values during training, which differs from the positional cues provided by the x and y axes. In contrast, information is represented equally for all axes with the explicit supervision, which results in similar variance for all positional parameters $t_{1,2,3}$.

Because the rotation parameters are ambiguous (i.e., different parameterizations result in the same outcome), parameter predictions cannot be compared directly to the ground truth values. IoU values, which capture the overall prediction performance, are, therefore, also reported in Table 1. The distributions of IoU scores over the test dataset for both learning strategies are shown in Fig. 8. As expected, the supervised approach results in a better parameter estimation overall, since labeled parameters are provided during training and the 3D appearance of the target superquadrics is reconstructed very closely from the predicted parameters. The model trained with explicit supervision reaches an IoU score of 94.6% on test depth images containing a single superquadric. The model learned with implicit supervision, on the other hand, performs somewhat weaker with an IoU score of 85.6% and a slightly larger variance. Nevertheless, given that no ground truth parameters are needed to learn the predictor with implicit supervision, this learning strategy may be better suited for more challenging problems involving real world data, where visual appearance is the only factor available to drive the learning procedure.

TABLE 2. Comparison with the state-of-the-art. We compare our models to the approach proposed by Solina and Bajcsy [9] and Paschalidou *et al.* [19] on the whole test dataset (with 20k images), as well as on a subset (with 8.93k images), which represents the intersection of the superquadric vocabulary considered in all papers.

Test dataset	Method	IoU [%]
full (20k) $(0.1 \leq \epsilon_{1,2} \leq 1)$	Explicit sup. $\mathcal{L}_{OC}$ (ours)	$\mathbf{94.62 \pm 3.18}$
	Implicit sup. $\mathcal{L}_{DR}$ (ours)	85.64 ± 5.72
	Solina and Bajcsy [9]	84.51 ± 8.89
	Paschalidou *et al.* [19]	81.48 ± 6.67
subset (8.93k) $(0.4 \leq \epsilon_{1,2} \leq 1)$	Explicit sup. $\mathcal{L}_{OC}$ (ours)	$\mathbf{95.52 \pm 2.65}$
	Implicit sup. $\mathcal{L}_{DR}$ (ours)	86.88 ± 5.16
	Solina and Bajcsy [9]	82.58 ± 9.28
	Paschalidou *et al.* [19]	83.29 ± 6.15

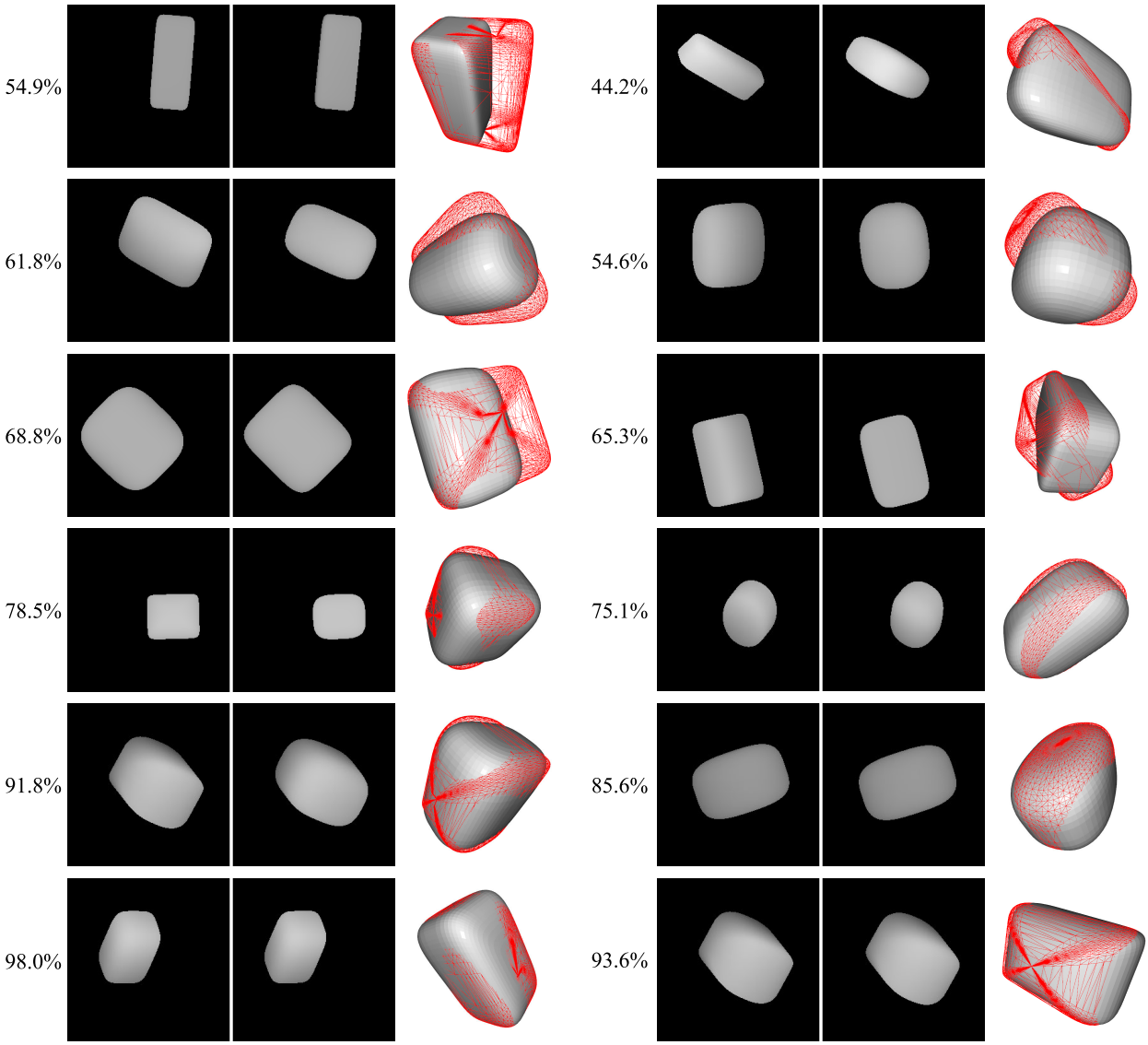
2) COMPARISON WITH THE STATE OF THE ART

The next series of experiments compares the proposed models to the state-of-the-art in superquadric recovery. Specifically, results are compared with the iterative minimization method from Solina and Bajcsy [9] and the more recent work (also using CNNs) from Paschalidou *et al.* from [19].

To ensure a fair comparison to the work of Paschalidou *et al.* [19] we use the source code provided by the authors.[1] Because the solution from [19] was proposed for the recovery of multiple superquadrics, we limit the output of their model to a single superquadric. The model was originally trained on ShapeNet objects represented in voxel form, so we retrain it on our superquadric dataset while discarding the parsimony loss. The loss is not needed for our use case as it only helps to estimate the number of superquadrics present in the data. Instead of using depth images as input, the voxel representation of the superquadrics is used with the same grid size of $128 \times 128 \times 128$ as advocated in [19]. This voxel representation gives Paschalidou *et al.* a slight advantage over the models proposed in this work, since depth images only contain object points from the objects front-face relative to the viewing angle, while the back-face is occluded. This is not the case with the voxel representation. As the source code provided by [19] was used for the experiments, where the shape parameter predictions are hard-coded between 0.4 and 1.5, we present results on the our full dataset with the ϵ values ranging between 0.1 and 1 as well as on its subset containing only superquadrics whose shape parameters fall into the range of intersection of both papers, namely between 0.4 and 1. In the same fashion as [19] we train the model on the entire training set for $40k$ iterations. We also use the source code provided by [9] for the iterative minimization method. No modification is needed for this approach, which is used directly with the images in our dataset.

The results of the comparison in terms of average IoU scores and standard deviations are shown in Table 2. On the whole test dataset, the model from [19] achieves a smaller IoU score in comparison to our supervised and unsupervised models by 13.14% and 4.16%, respectively.

[1] https://github.com/paschalidoud/superquadric_parsing



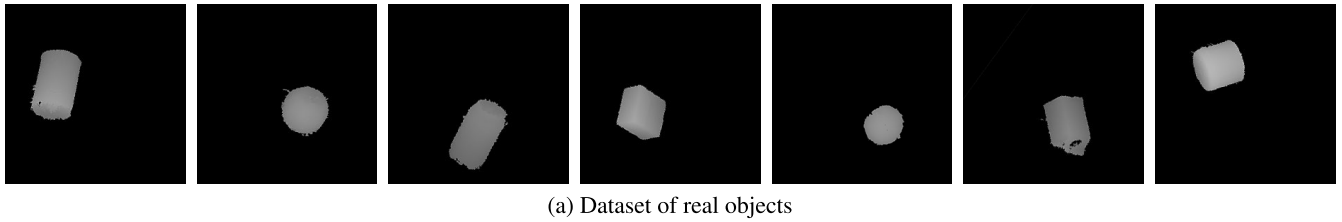
(a) Explicit supervision $\mathcal{L}_{OC}$ (b) Implicit supervision $\mathcal{L}_{DR}$

FIGURE 9. Qualitative results. We show examples of both models across the whole IoU distribution range. For each example, from left to right, we show: the ground-truth depth image, the reconstructed depth image, and a 3D render of the predicted (gray) and ground-truth (red) superquadrics. The figure is best viewed electronically.

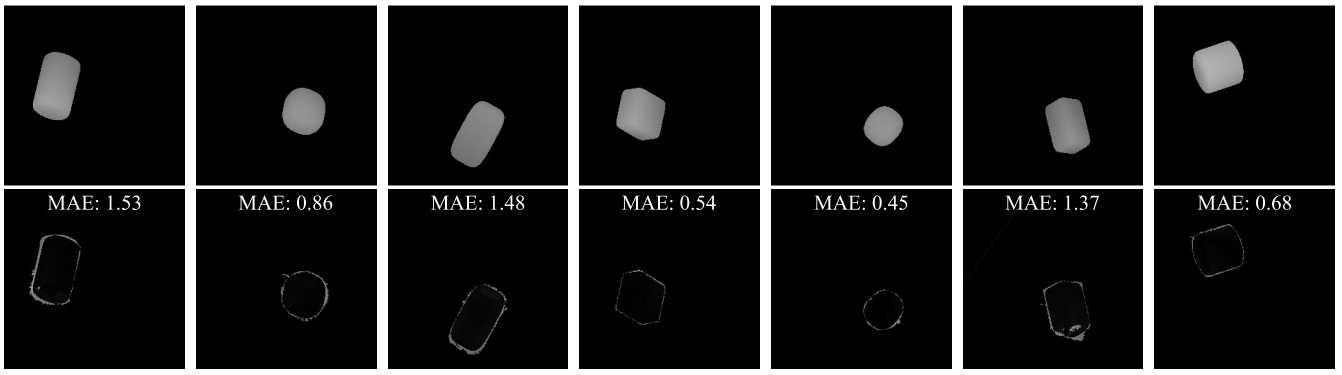
The comparison on the full dataset works in our favor, since the model of [19] is not capable of predicting superquadrics with sharp edges ($\epsilon_{1,2} < 0.4$). The comparison on the subset of the test dataset shows a similar picture with our models maintaining higher IoU accuracies. While the model from [19] does show the biggest improvement of 1.81% on the subset of examples, our models also perform better (by 0.9% and 1.24% IoU). Note, that the method from Paschalidou *et al.* is originally designed to predict multiple superquadrics as individual parts of a more complex object. Nevertheless, we can observe a better performance of our models on the base case – the recovery of a single superquadric.

When looking at the results for the approach from Solina and Bajcsy [9], it can be seen that both proposed models also outperform the iterative approach. While the model learned with implicit supervision only results in slightly higher IoU values on both the full test set as well as the smaller subset of images, the model trained with explicit supervision achieves significantly better results comparatively. Additionally, the variance of the scores is considerably smaller with both proposed models than the variance of the scores produced by the solution from [9].

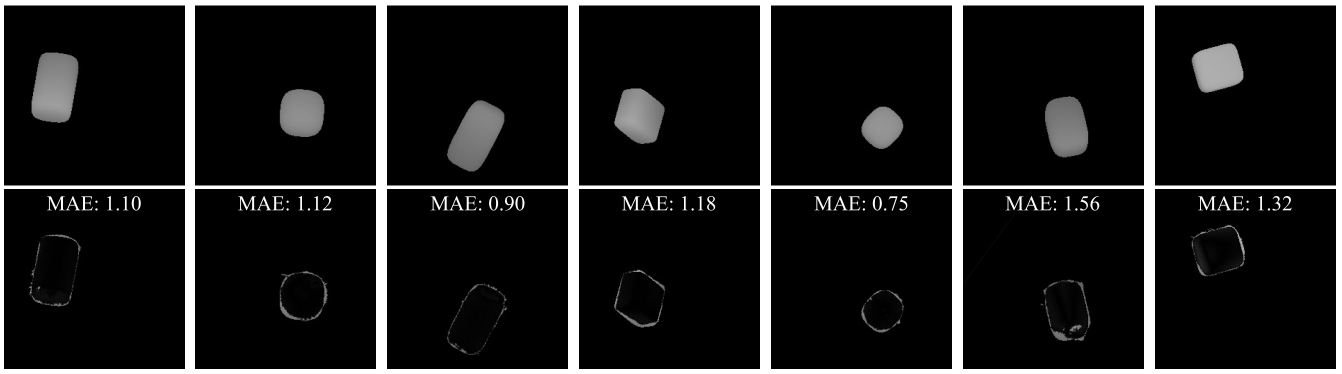
In Table 3, we observe the average processing times, required by each approach. With our CNN predictors, we are able to process an image in 3 ms on average. In comparison, the model from Paschalidou *et al.* [19] requires 1 ms and the iterative method [9] requires 690.52 ms on average. All CNN models are able to process the data in a single forward pass through the network, which results in such a drastic



(a) Dataset of real objects



(b) Explicit supervision $\mathcal{L}_{OC}$



(c) Implicit supervision $\mathcal{L}_{DR}$

FIGURE 10. **Qualitative results on real data. We show how our models work on (a) a collection of real-world data, captured with a 3D scanner. The results can be observed for (b) the explicitly supervised model and for (c) the implicitly supervised model. For (b) and (c), the top row represents the reconstructed superquadric and the bottom row shows the absolute difference between images along with the Mean Absolute Error (MAE).**

TABLE 3. **Processing time comparison. We compare the processing time of our models to the iterative method by Solina and Bajcsy [9] and Paschalidou *et al.* [19] on the whole test dataset. Rounded average processing time computed over all test data is reported.**

Method	Processing time
Explicit supervision $\mathcal{L}_{OC}$ (ours)	3 ms
Implicit supervision $\mathcal{L}_{DR}$ (ours)	3 ms
Solina and Bajcsy [9]	690 ms
Paschalidou *et al.* [19]	**1** ms

decrease in processing time in comparison to the classic iterative method. Paschalidou *et al.* [19] achieve a slightly better performance in comparison to our models, since the authors use an architecture with 5 convolutional layers in comparison to ours, which has 18 convolutional layers to process the depth images.

3) QUALITATIVE ANALYSIS

Next, we present a few qualitative examples of the fitting result produced by the proposed two models. A selection of examples with various IoU scores is shown in Fig. 9. We first observe the model trained with explicit supervision in the left column. Examples scoring less than 70% IoU are usually cases where the amount of self-occlusion is the highest and only a single side of the superquadric is visible, similar to those in the first and third row. Consequently, the model has to guess the size of the superquadric along the occluded axis, based on similar examples from training data. Such cases are expected, since the occluded data is lost at the moment of capture. We observe that even in examples with lower IoU (below 70%), the rotation is correctly estimated at least in the major axis of the superquadric. Among the results with 70% IoU only corner roundness is slightly overestimated.

In the right column are predictions, made by the model trained with implicit supervision. Examples with lower IoU scores (below 70% in the example) have degenerated properties, e.g., the contour of superquadrics matches closely the ground truth depth images, however, the actual 3D shape is not fitted properly. In cases with high IoU scores (80% and

above), the error does not come from shape parameters as with the explicitly learned model, but instead comes from the slight displacement of the superquadric. This might be due to the small difference between the original depth images and images, reconstructed by our differentiable renderer.

4) PERFORMANCE ON REAL DATA

Finally, we analyze the performance of the two models on real data. The Artec MHT 3D scanner is used to capture point cloud data of real objects. The point clouds are then transformed into meshes and outlier points are removed. A depth image of the mesh is created by reading from the depth buffer. The scanned objects are captured in fixed rotation. Finally the captured dataset is augmented by adding a random rotation to each object.

We use both models to recover parameters from real images and visualize the results in Fig. 10. Overall, we observe a tight fitting of the predicted superquadrics on the scanned objects. With every example, the volume of the superquadric is slightly underestimated. Both models appear to be handling well the noise around the borders of the scanned objects. The models are able to interpret the different shapes of the objects. For example, they are able to differentiate between a food can with sharp edges versus a drink can with rounded edges. They are also insensitive to some amount of deformation, which is demonstrated with the object, shown second from the right. In general, the explicitly learned model performs better and results in an average MAE of 0.98, while in contrast, the implicitly learned model achieves an average MAE of 1.13.

V. CONCLUSION

The paper addressed the problem of superquadric recovery from depth images and presented two strategies for learning CNN models capable of predicting the parameters of a single superquadric in depth data. The proposed learning strategies rely on loss functions that capture the geometric properties of superquadrics in general position and penalize the shape, size, translation and rotation parameters. The first proposed strategy uses explicit supervision and compares the predicted superquadric parameters with the ground-truth parameters by reconstructing 3D occupancy voxel grids. The second strategy uses implicit supervision and relies on appearance comparisons. Specifically, it reconstructs a depth image from the predicted parameters and compares it to the input depth image. Both methods surpass the classic iterative method from [9] and the more recent deep learning method from [19] in terms of IoU accuracy on our experimental dataset.

By successfully improving the process of parameter recovery of a single superquadric, we showed that the accuracy of 3D reconstruction for objects modelled by a single superquadric can be improved in comparison to existing techniques. This may lead to better performing (biologically relevant) solutions for complex reconstruction problems, where recovery of multiple superquadrics is needed. In addition, the model trained with implicit supervision offers the possibility of training on depth images only (without ground-truth parameters), which alleviates the need for labeled training data. We also demonstrated the ability to recover superquadric parameters from depth images of real-world objects by learning only from synthetic data.

However, to compete with more complex methods, we need to extend our learning strategies to support simultaneous segmentation and recovery of multiple superquadrics. We believe that the directions, outlined in this article, have potential for extensions that would allow for the interpretation of more complex objects and environments, where more than a single superquadric model is needed to describe the scene.

REFERENCES

[1] P. Khosla and R. Volpe, "Superquadric artificial potentials for obstacle avoidance and approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1998, pp. 1778–1784.
[2] N. E. Smith, R. G. Cobb, and W. P. Baker, "Incorporating stochastics into optimal collision avoidance problems using superquadrics," *J. Air Transp.*, vol. 28, no. 2, pp. 65–69, Apr. 2020.
[3] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp planning via decomposition trees," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 4679–4684.
[4] G. Vezzani, U. Pattacini, and L. Natale, "A grasping approach based on superquadric models," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1579–1586.
[5] I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psychol. Rev.*, vol. 94, p. 115, Apr. 1987.
[6] A. P. Pentland, "Perceptual organization and the representation of natural form," in *Readings Computer Vision*. Amsterdam, The Netherlands: Elsevier, 1987, pp. 680–699.
[7] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. New York, NY, USA: Henry Holt and Co, 1982.
[8] A. H. Barr, "Superquadrics and angle-preserving transformations," *IEEE Comput. Graph. Appl.*, vol. 1, no. 1, pp. 11–23, Jan. 1981.
[9] F. Solina and R. Bajcsy, "Recovery of parametric models from range images: The case for superquadrics with global deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 2, pp. 131–147, 1990.
[10] T. E. Boult and A. D. Gross, "Recovery of superquadrics from 3-D information," in *Proc. Intell. Robot. Comput. Vis.*, Feb. 1988, pp. 128–137.
[11] S. Voisin, M. A. Abidi, S. Foufou, and F. Truchetet, "Genetic algorithms for 3d reconstruction with supershapes," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 529–532.
[12] A. Leonardis, A. Jaklic, and F. Solina, "Superquadrics for segmenting and modeling range data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 11, pp. 1289–1295, Dec. 1997.
[13] Y. Yamane, E. T. Carlson, K. C. Bowman, Z. Wang, and C. E. Connor, "A neural code for three-dimensional object shape in macaque inferotemporal cortex," *Nature Neurosci.*, vol. 11, no. 11, pp. 1352–1360, Nov. 2008.
[14] J. A. Mazer, "So many pixels, so little time," *Nature Neurosci.*, vol. 11, no. 11, pp. 1243–1244, Nov. 2008.
[15] J. J. DiCarlo, D. Zoccolan, and N. C. Rust, "How does the brain solve visual object recognition?" *Neuron*, vol. 73, no. 3, pp. 415–434, Feb. 2012.
[16] D. L. K. Yamins and J. J. DiCarlo, "Using goal-driven deep learning models to understand sensory cortex," *Nature Neurosci.*, vol. 19, no. 3, pp. 356–365, Mar. 2016.
[17] R. D. Freeman, "Stereoscopic vision: Which parts of the brain are involved?" *Current Biol.*, vol. 9, pp. 610–613, 1999.
[18] S. Tulsiani, H. Su, L. J. Guibas, A. A. Efros, and J. Malik, "Learning shape abstractions by assembling volumetric primitives," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2635–2643.
[19] D. Paschalidou, A. O. Ulusoy, and A. Geiger, "Superquadrics revisited: Learning 3D shape parsing beyond cuboids," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, p. 10.
[20] D. Paschalidou, L. Van Gool, and A. Geiger, "Learning unsupervised hierarchical part decomposition of 3D objects from a single RGB image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1060–1070.

[21] T. Oblak, K. Grm, A. Jaklic, P. Peer, V. Struc, and F. Solina, "Recovery of superquadrics from range images using deep learning: A preliminary study," in *Proc. IEEE Int. Work Conf. Bioinspired Intell. (IWOBI)*, Jul. 2019, pp. 45–52.

[22] A. P. Pentland, "Recognition by parts," in *Proc. Int. Conf. Comput. Vis.*, 1987, pp. 612–620.

[23] F. Solina and R. Bajcsy, "Range image interpretation of mail pieces with superquadrics," in *Proc. Conf. Artif. Intell.*, 1987, pp. 733–737.

[24] A. D. Gross and T. E. Boult, "Error of fit measures for recovering parametric solids," in *Proc. 2nd Int. Conf. Comput. Vis.*, 1988, pp. 690–694.

[25] F. P. Ferrie, J. Lagarde, and P. Whaite, "Darboux frames, snakes, and superquadrics: Geometry from the bottom up," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 8, pp. 771–784, 1993.

[26] T. Horikoshi and S. Suzuki, "3D parts decomposition from sparse range data using information criterion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 0, pp. 168–173.

[27] K. Wu and Levine, "Recovering parametric geons from multiview range data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 159–166.

[28] A. Jaklič, A. Leonardis, and F. Solina, *Segmentation Recovery Superquadrics*. Norwell, MA, USA: Kluwer, 2000.

[29] N. Vaskevicius and A. Birk, "Revisiting superquadric fitting: A numerically stable formulation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 220–233, Jan. 2019.

[30] A. J. Hanson, "Hyperquadrics: Smoothly deformable shapes with convex polyhedral bounds," *Comput. Vis., Graph., Image Process.*, vol. 44, no. 2, pp. 191–210, Nov. 1988.

[31] D. Terzopoulos and D. Metaxas, "Dynamic 3D models with local and global deformations: Deformable superquadrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 7, pp. 703–714, Jul. 1991.

[32] L. Chevalier, F. Jaillet, and A. Baskurt, "Segmentation and superquadric modeling of 3D objects," *J. Winter School Comput. Graph.*, vol. 11, no. 1, p. 8, Feb. 2003.

[33] J. Šircelj, T. Oblak, K. Grm, U. Petkovič, A. Jaklič, P. Peer, V. Štruc, and F. Solina, "Segmentation and recovery of superquadric models using convolutional neural networks," in *Proc. Comput. Vis. Winter Workshop*, 2020, pp. 1–5.

[34] J. Slabanja, B. Meden, P. Peer, A. Jaklic, and F. Solina, "Segmentation and reconstruction of 3D models from a point cloud with deep neural networks," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2018, pp. 118–123.

[35] A. Jakliä, M. Eri, I. Mihajlovi, Ž. Stopináek, and F. Solina, "Volumetric models from 3D point clouds: The case study of sarcophagi cargo from a 2nd/3rd century AD roman shipwreck near sutivan on island Braä, croatia," *J. Archaeol. Sci.*, vol. 62, pp. 143–152, Oct. 2015.

[36] V. Z. Stopinšek and F. Solina, "3D modeliranje podvodnih posnetkov," in *Proc. Si Robot.*, Slovenska, Bratislava, 2017, pp. 103–114.

[37] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[38] J. Wu, Y. Wang, T. Xue, X. Sun, B. Freeman, and J. Tenenbaum, "MarrNet: 3D shape reconstruction via 2.5D sketches," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 540–550.

[39] S. Liu, L. Giles, and A. Ororbia, "Learning a hierarchical latent-variable model of 3D shapes," in *Proc. Int. Conf. 3D Vis.*, Sep. 2018, pp. 542–551.

[40] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 165–174.

[41] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4460–4470.

[42] S. Miao, Z. J. Wang, and R. Liao, "A CNN regression approach for real-time 2D/3D registration," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1352–1363, May 2016.

[43] R. Zhu, H. K. Galoogahi, C. Wang, and S. Lucey, "Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 57–65.

[44] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Proc. 14th Robot., Sci. Syst.*, Dec. 2018.

[45] J. B. Kuipers, *Quaternions and Rotation Sequences: A Primer With Applications to Orbits, Aerospace, and Virtual Reality*. Princeton, NJ, USA: Princeton Univ. Press, 1999.

[46] M. Gadelha, R. Wang, and S. Maji, "Shape reconstruction using differentiable projections and deep priors," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 22–30.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[48] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[49] K. Shoemake, "Uniform random rotations," in *Graphics Gems III*. Amsterdam, The Netherlands: Elsevier, 1992, pp. 124–132.

[50] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic gradient descent," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–5.

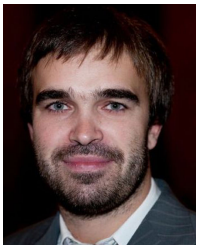
[51] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Proc. Neural Netw., Tricks trade*, 2012, pp. 437–478.

[52] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, "On large-batch training for deep learning: Generalization gap and sharp minima," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–8.

[53] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Exp.*, vol. 6, no. 4, pp. 312–315, Dec. 2020.

TIM OBLAK received the double M.Sc. degree from the University of Ljubljana, Slovenia, and the Graz University of Technology, in 2020. He is currently pursuing the Ph.D. degree with the University of Ljubljana. During his master's studies at the Faculty of Computer and Information Science in Ljubljana, he worked on the subject of 3D reconstruction, which eventually culminated in his master's thesis. His Ph.D. topic deals with computer vision in the domain of image-based biometrics. He is currently a Teaching Assistant with the University of Ljubljana.

JAKA ŠIRCELJ (Member, IEEE) received the bachelor's degree in physics from the Faculty of Mathematics and Physics, Ljubljana, and the master's degree from the Faculty of Computer and Information Science, Ljubljana. He is currently a Researcher with the University of Ljubljana, Slovenia, where he works on 3D object reconstruction using deep learning. His research interests include computer vision, 3D vision, and in machine learning security, namely in adversarial learning.



VITOMIR ŠTRUC (Senior Member, IEEE) received the Ph.D. degree from the Faculty of Electrical Engineering, Ljubljana, in 2010. He is currently an Associate Professor with the University of Ljubljana, Slovenia. His research interests include problems related to biometrics, computer vision, image processing, pattern recognition, and machine learning. He has (co)authored more than 100 research papers for leading international peer reviewed journals and conferences in these and related areas. He is a member of IAPR, EURASIP, and Slovenia's national contact point for the EAB, and the current President of the Slovenian Pattern Recognition Society (Slovenian Branch of IAPR). He has served in different capacities on the organizing committees for several top-tier vision conferences, including IEEE Face and Gesture, ICB, WACV, and others. He has served as the Area Chair for WACV 2018, 2019, 2020, ICPR 2018, Eusipco 2019, and FG 2020. He is the Program Co-Chair of the 2020 International Joint Conference on Biometrics (IJCB). He is also a Senior Area Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY; and an Associate Editor of *Pattern Recognition*, *Signal Processing*, and *IET Biometrics*.

PETER PEER (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Ljubljana, Slovenia, in 2003. He is currently a Professor of computer science with the University of Ljubljana, where he heads the Computer Vision Laboratory, coordinates the double degree study program with the Kyungpook National University, South Korea, and serves as the Vice-Dean for economic affairs. Within his post-doctorate, he was an Invited Researcher at CEIT, Donostia-San Sebastian, Spain. He has participated in several national and EU funded research and development projects and published more than 100 research papers in leading international peer reviewed journals and conferences. His research interests include biometrics, color constancy, image segmentation, detection, recognition, and real-time computer vision applications. He is a member of EAB and IAPR. He served as the Chairman for the Slovenian IEEE Computer Chapter for four years. He is the Co-Organizer of the Unconstrained Ear Recognition Challenge and Sclera Segmentation Benchmarking Competition. He serves as an Associate Editor for IEEE Access and *IET Biometrics*.

FRANC SOLINA (Life Senior Member, IEEE) received the Dipl.Ing. and M.S. degrees in electrical engineering from the University of Ljubljana, Slovenia, in 1979 and 1982, respectively, and the Ph.D. degree in computer and information science from the University of Pennsylvania, in 1987. In 1988, he started teaching at the Faculty of Computer and Information Science, University of Ljubljana. Since 2011, he has also in the Video and New Media Program at the Academy of Fine Arts and Design, University of Ljubljana. In 1991, he founded the Computer Vision Laboratory, Faculty of Computer and Information Science. From 2006 to 2010, he has served as the Dean of the Faculty of Computer and Information Science. He is currently a Professor of computer science with the University of Ljubljana. His main research interests include 3D modeling from images and the use of computer vision in human–computer interaction, heritage science, and in art installations. He has (co)authored more than 200 research papers for international peer reviewed journals and conferences in these and related areas. He is a Fellow of IAPR, a member of ICOMOS and Slovenian Association of Fine Arts Societies, and a regular member of the European Academy of Sciences and Arts, Salzburg.

ALEŠ JAKLIČ (Member, IEEE) received the Ph.D. degree in computer and information science from the University of Ljubljana, Slovenia, in 1997. He is currently an Assistant Professor of computer science with the University of Ljubljana. In 1997, he was elected into Partridge Fellowship at the Fitzwilliam College, University of Cambridge, U.K. His research interests include 3D image segmentation, computer vision applications, computing education, and the IoT.