Segmentation and Reconstruction of 3D Models from a Point Cloud with Deep Neural Networks

Jurij Slabanja, Blaž Meden, Peter Peer, Aleš Jaklič, Franc Solina
Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia
E-mail: jurijslabanja@gmail.com, {blaz.meden,peter.peer,ales.jaklic,franc.solina}@fri.uni-lj.si

***Abstract*—The need to model visual information with compact representations has existed since the early days of computer vision. We implemented in the past a segmentation and model recovery method for range images which is unfortunately too slow for current size of 3D point clouds and type of applications. Recently, neural networks have become the popular choice for quick and effective processing of visual data. In this article we demonstrate that with a convolutional neural network we could achieve comparable results, that is to determine and model all objects in a given 3D point cloud scene. We started off with a simple architecture that could predict the parameters of a single object in a scene. Then we expanded it with an architecture similar to Faster R-CNN, that could predict the parameters for any number of objects in a scene. The results of the initial neural network were satisfactory. The second network, that performed also segmentation, still gave decent results comparable to the original method, but compared to the initial one, performed somewhat worse. Results, however, are encouraging but further experiments are needed to build CNNs that will be able to replace the state-of-the-art method.**

***Index Terms*—computer vision, segmentation, 3D reconstruction, superquadrics, point cloud, deep neural networks, TensorFlow, Keras**

I. Introduction

How can the 3D structure of the visually perceived environment be represented by a computer? For a robot, it is important that this information is represented with 3D models, making it easy for it to navigate and interact with a constantly changing environment. Applications that require such representation range from autonomous driving to robot grasping. Up until now, one of the standard mathematical 3D models for such representation on a part-level abstraction were superquadrics. Superquadrics are shape primitives out of which more complex shapes can be built. With the same parametrisation, superquadrics can take up the shape of several common geometric bodies (i.e. cuboids, cylinders and ellipsoids). But the state-of-the-art method for reconstruction of superquadrics from 3D points requires an iterative minimisation of a non-linear fitting function and it is consequently much too slow for most current computer vision applications. Another processing bottleneck is caused by the abundance of 3D data acquired by a new generation of 3D sensors and methods, ranging from Microsoft Kinect to multi-image photogrammetry. The recent success of Convolutional Neural Network (CNN) methodology opens up a completely new venue to solve the recovery of superquadrics from 3D data in real-time.

In this article we will therefore try to demonstrate, using standard CNN approaches, that the following problems can be solved:

1) Development of a CNN for model recovery of a single superquadric. Such CNN should be already directly applicable for robotic manipulation where data acquisition often pre-empts the need for segmentation of 3D point clouds.
2) Development of a CNN for concurrent segmentation and model recovery of several superquadrics. We will embed the CNN solution from recovery of single superquadrics into a larger deep model and develop a training procedure around the concept of multi-task learning. This CNN would in principle reproduce the functionality of the state-of-the-art method but at a much higher speed and consequently for a much larger data input capacity, making the new solution relevant for many new application domains, particularly for applying robots in unconstrained environments.

The rest of the article is divided into sections as follows: first we describe the part-level models, i.e. superquadrics, and the state-of-the-art methods of their recovery. Next, we describe for which applications are superquadric models typically used for, and the present state of how CNNs are used for processing of 3D visual data. Finally, we describe two experiments that we made (i) development of a CNN for recovery of single superquadric models and (ii) development of a CNN for concurrent recovery of several superquadrics. The article concludes with a discussion of this first attempt of using CNNs for recovery of superquadrics from 3D data.

II. Superquadrics as part-level 3D shape models

The mathematical models for perceptual blocks should represent a wide variety of most common 3D shapes and most importantly, a reliable method should exist for their recovery directly from images. Alex Pentland [22] proposed superquadrics to fill this role of visual blocks due to their unique parametrisation of a large family of 3D shapes (Fig. 1). Superquadrics help in reducing unessential bits of information and enable a higher level of abstraction. We studied how to determine the proper scale for modelling visual data since the nature of a scene and the visual goals should determine the appropriate scale of observations [29]. The expressive power of superquadrics can be additionally enlarged by applying global parametric deformations such as bending and tapering

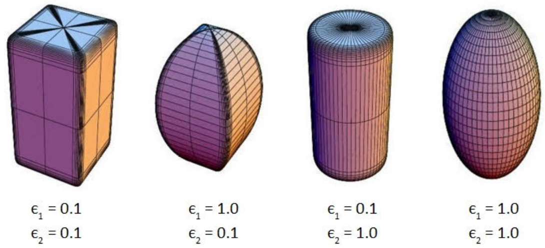

Fig. 1. Superquadrics are part-level blocks used in computer vision to assemble more complex shapes. Here are shown four superquadrics of the same size but with different values of shape parameters ϵ_1 and ϵ_2 (see Eq. 1). The advantage of superquadrics is that several common geometric bodies (i.e. cuboids, cylinders and ellipsoids) can all be represented with the same parametrisation.

[15] or local deformations as an additional layer of details. Superquadrics can be covered also with photographic texture since the combination of 3D shape with appearance features empowers superior recognition—how else to distinguish an ellipsoid shape as a melon or a coconut? Due to their symmetrical shape, superquadrics predict also the self-occluded side of an object, when the object is seen just from a single view-point. Despite some initial competition with generalized cylinders [23], superquadrics prevailed in the long run.

III. The state-of-the-art method of superquadric recovery

Pentland's proposal to recover superquadric from shading information in 2D intensity images was overly complicated and not successful in practice. We decided to use as input explicit 3D information in the form of range images [28]. Range images are a uniform 2D grid of 3D points as seen from a particular viewpoint.

We were the first to formulate a working method for recovery and segmentation of superquadrics from range images, which we reported in a series of articles [17], [28], culminating in a research monograph [15]. First, we modified function F:

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

to fit a superquadric model to 3D points [28], where a_1, a_2, and a_3 represent the size of the superquadric model, while ϵ_1 and ϵ_2 determine its shape (see Fig. 1). All points x, y and z for which $F(x,y,z) = 1$ lie on the surface of the superquadric. Later, we expanded this method, to simultaneously deconstruct the input range image into several superquadrics, resulting in a perceptually relevant segmentation [17]. Since the fitting function is highly non-linear and the segmentation process is gradually progressing, an iterative procedure could not be avoided. At the beginning of the process, superquadric seeds are overlayed on 3D points which can expand in a controlled fashion, as directed by the distribution of 3D points in their vicinity; At intermediate steps in the superquadric growth process, superfluous/overlapping superquadrics are eliminated to arrive to the final result where for each perceptual part a single superquadric model is recovered. In this way any given scene can be segmented only into such parts, that can be modelled with superquadrics.

IV. What are superquadrics used for?

Our method of superquadric recovery or derivatives of it are used for a variety of applications, at present mostly in robotics for path and grasp planning [34]. The most recent research citing our method, published in 2018, focused on 3D object recognition in RGB-D images where objects need to be recognised either for the purpose of being grasped and manipulated by a robot, or where the entire scene must be recognised to allow high-level cognitive tasks to be performed [30]. Part-level description is essential for grasping and handling ob objects of various shapes which Internet retailers and shipping companies strive to automate at this time [18].

As an example of recent successful use of superquadric models in a specific application domain serves our collaboration with underwater archeologists in 3D documentation of archeological sites and artifacts [7]. We resurrected our superquadric related research to model stone blocks and sarcophagi on a 2nd/3rd century AD Roman shipwreck near Sutivan on island Brač, Croatia [14] and deformed superquadrics for amphorae [31] from dense 3D point clouds. Such models can simplify the computation of the volume of the ship's cargo and hence it's tonnage and size and aid the archeological interpretation. Cameras mounted on autonomous flying drones or other autonomous vehicles can via multi-image photogrammetry quickly collect image data in the form of dense 3D point clouds over large areas [1]. LiDAR is another recent technology that can collect huge amounts of 3D points from the air [37]. The micro structure of the collected 3D data exhibited as 3D points should be for any interpretation aggregated into their perceptually relevant macro structure, that is to clearly delineate what objects or their parts are contained in the data. There are several standard approaches to segmentation of 3D point clouds, such as the popular RANSAC method, that was used, for example, for segmenting LiDAR data into cylindrical shapes [25]. Recently, symmetry as one of the gestalt cues that helps in organising visual information, was used as a grouping principle to segment objects in 3D point clouds of cluttered scenes [3]. By their inherent symmetry, superquadrics also detect and anticipate symmetrical shapes even for the self-occluded parts of a scene.

Owing to this collaboration in the digital heritage domain we realised that part-level modelling of 3D point clouds is very much in demand, especially in the light of huge advances and increased capabilities of capturing 3D data, and that our superquadric recovery method is not dated. Using this method, one can localise individual parts in large 3D point clouds or search for specific shapes. At the same time, however, we became fully aware of the *major drawback of the state-of-the-art method of superquadric recovery, which is the slow speed* due to the iterative nature of the process, making it unsuitable for practically any real-time applications, such as the now all-consuming problem of autonomous driving [21].

V. CNNs and 3D data

Deep learning methods using CNNs have become an omnipresent and highly successful part of recent approaches in imaging and vision [16]. A CNN consists of several inter-connected layers between the input and output layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers. CNN based computational approach in computer vision is in general very fast, can cope with large data input, and has also similarities with the way how our brains are coping with processing of visual data. The key to successful CNN solutions is to select a proper network architecture and to set the optimal values of various internal parameters of the network. This requires a learning phase where typically a huge number of examples, together with the correct results, must be presented to the network. In general, the more training examples, the more accurate results one can expect. The majority of research involving deep neural networks uses as input 2D intensity or colour images to arrive at the output to some identification or classification of the input image [26].

Like most computer vision research groups these days, we are collecting experience in CNN research related to 2D image data [5], [6], [10], [19]. Inspired by the success of the whole CNN paradigm we realised that segmentation and recovery of superquadrics could be solved much faster under this new framework. In this project, however, we are entering the 3D domain which is acutely underrepresented in the current CNN research, although CNNs have been already employed to process 3D visual data and use 3D models. A CNN regression approach was used for real-time 2D/3D registration which was, as superquadric recovery up till now, traditionally solved by iterative computation [20]. CNNs have been used to estimate the viewpoint direction of a 3D scene [32]. There has already been some work on recovering volumetric models using deep neural networks [9], [27], [35]. Wu et al. [35] are building voxel representations of objects, called 3D shapenets, from range images and use CNNs to complete the shape, determine the next best view, and to recognise objects. Sharma et al. [27] extended shapenets into full convolutional volumetric auto encoder by estimating voxel occupancy grids. Grant et al. [9] use CNNs to predict volumes on previously unseen image data. 3D human pose estimation using CNNs can be done from monocular video [38]. Much research effort related to CNNs is directed towards faces and the use of 3D information in this context. CNNs are used to estimate face normals instead of standard shape-from-shading methods [33] or for fitting 3D morphable models to faces captured in unconstrained conditions [2], [8], [11], [13]. Object level segmentation of vehicles from urban LiDAR scans using CNNs has been studied [36]. Large scale LiDAR scans are usually too large to apply directly standard CNN models. Downscaling such large scans, on the other hand, can impact the segmentation and recognition task. Processing of very large sets of 3D image data with CNNs would require some intelligent subdivision and parallelization of the task. There have already been studies on using both intensity images and depth images in a multimodal deep learning scenario for object recognition [4].

There is therefore ample evidence by current research that the marriage of 3D data and models with CNN computational paradigm is promising but only starting. To our knowledge, no method exists yet for recovery of part-level volumetric models, such as superquadrics from 3D point clouds using CNNs. Therefore, we decided to test this idea using some standard and pretrained CNN solutions.

VI. CNN-based approach to segmentation and model recovery from 3D point clouds

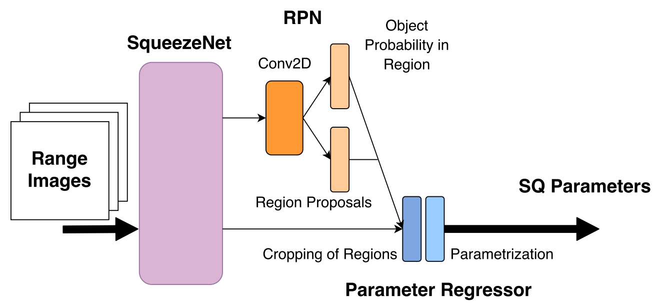

Fig. 2. Network architecture used in the described experiments. SqueezeNet is used as the base model and is connected to the Region Proposal Network (RPN) which determines possible object regions and Parameter Regressor network which outputs final predictions as bounding boxes.

A. CNN for recovery of single superquadrics

In the first experiment, we successfully estimated the model parameters of a single object using some standard CNN architectures. The database consisted of 100 training and 30 test scenes consisting of 3D points on the surface of a single superquadric. Each scene was represented with six orthogonal range images. The CNN network was trained in 12 epochs. Initially, the objects in the scene were spheres of different diameters. Thus the network had to determine just the radius of the sphere. With more complex and different shapes the accuracy of the parameter recovery dropped just a bit. We tested several different variations of CNN networks (dropout, weight regularisation, size of completely connected layers, different optimisation functions, etc.) and measured the accuracy of CNN results using the Mean Square Error among recovered and actual parameter values. Among different CNN architectures that we tested, the best performance was achieved with SqueezeNet [12].

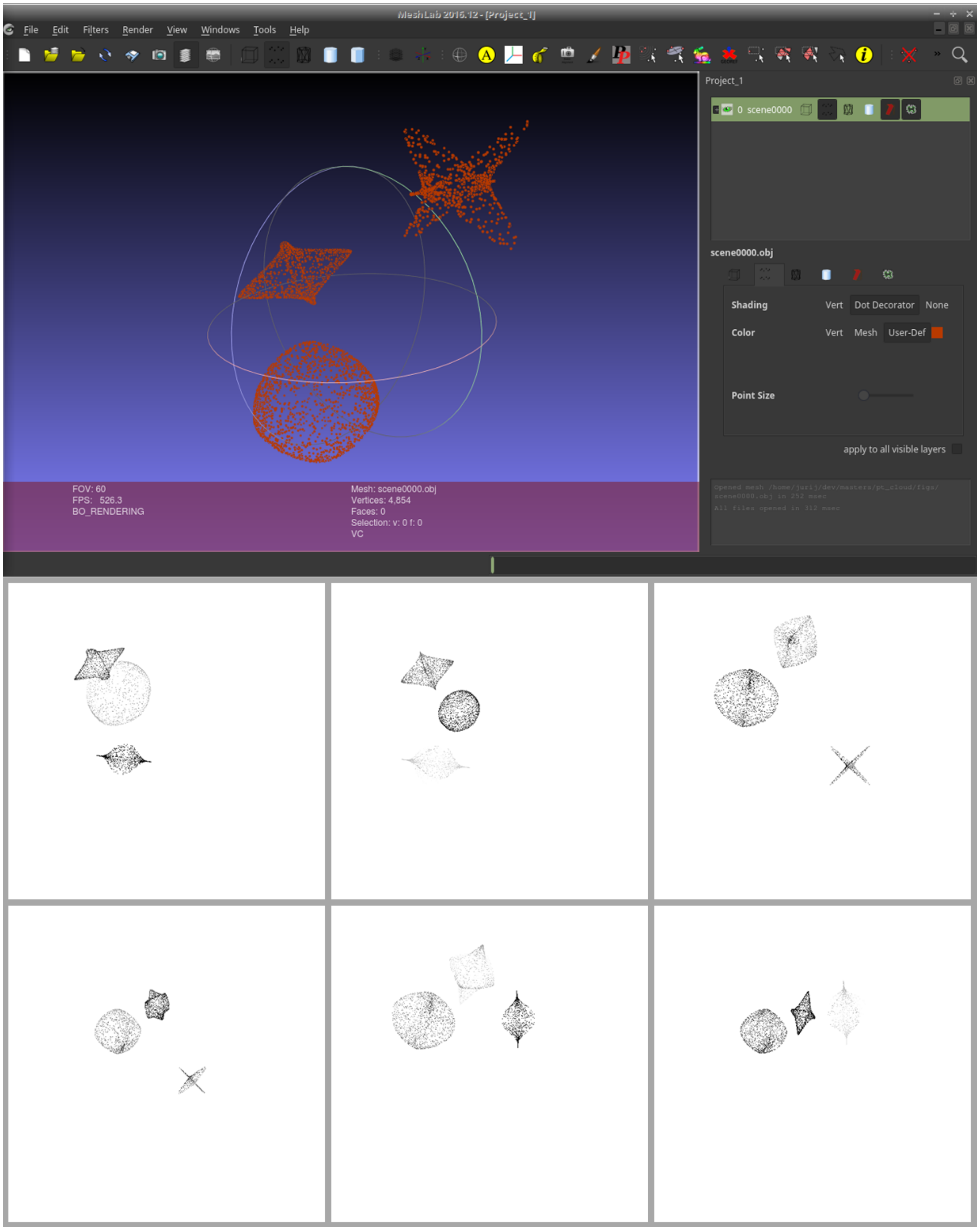

Fig. 3. On top is displayed a test scene for the second experiment, consisting of three 3D point clouds corresponding to three objects, visualized with MeshLab. At the bottom are six corresponding range images, generated from the six orthogonal views of the 3D point cloud.

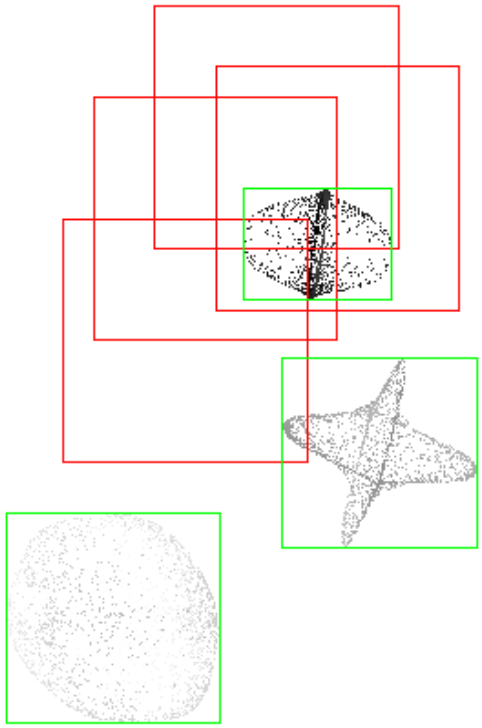

Fig. 4. Examples of regions, found with the Region Proposal Network (RPN) in the second part of the study. Green boxes indicate correctly segmented objects.

B. *CNN for concurrent segmentation and recovery of multiple superquadrics*

Next, we addressed also concurrent segmentation and superquadric recovery with a Region Proposal Network (RPN) and expanded the entire net into a Faster Region-based CNN [24] (Fig. 2). The database for this experiment consisted of 1000 training and 100 test scenes. For common layers we used the SqueezeNet architecture. Each scene was represented with six orthogonal range images into which three randomly positioned and differently shaped superquadrics were inserted (Fig. 3). In each training iteration we introduced to the network a single range image, to detect regions. Next, the regions were combined into batches of size 6, to train the regression network on all detected regions in batches. We used 10 epochs to train the network which took over 15 hours of processing time on a GTX 1080 Ti graphics card with 11GB video RAM. Training of the network in 10 epochs took on our equipment over 15 hours. Testing of the network on 100 scenes took 245s or just 0.4s for each range image, thus clearly illustrating that segmentation and recovery of superquadrics can also be performed in near real-time. In contrast to the state of the art solutions, the speed of a CNN solution is independent of the number of 3D points but only on the dimensions of the input range images. Fig. 4 shows an example scene where green boxes indicate correct locations of objects and red boxes indicate found regions for the topmost object. The implementation was based on Python, Keras and Tensorflow libraries.

VII. Discussion

This article describes the first attempt to use CNNs for recovery and segmentation of 3D models from 3D point clouds. As a 3D model we selected superquadrics. The unique superquadric parametrisation, allowing a uniform handling of most basic geometric 3D shapes (e.g. ellipsoids, cylinders, cuboids) is a big bonus for a CNN implementation which can effectively utilize this uniformity. The most important advantage over the state-of-the-art method of superquadric recovery is the real-time speed of the CNN solution. This can make superquadric modelling attractive also for other, new applications domains where the vision system has to provide information about the scene in real-time. One of such active and burning application domains is autonomous driving. Since this research is at a preliminary stage, we decided to perform tests initially with some publicly available standard CNN solutions.

In the first experiment, we focused on developing a CNN model for evaluating parameters of a single superquadric model from 3D data using the standard network architecture VGG16 (Fig. 2). In applications where the method of 3D point acquisition pre-empts any need for segmentation such a CNN for recovery of model parameters of a single supequadric would be quite sufficient. The method would have to be extended, however, to recover at the same time also the position and orientation parameters, so that the recovered model could be placed into a particular position in some world coordinate system. In the future, we will develop a CNN regressor trainable end-to-end that which will be able to regress the superquadric parameters directly from the input data. The key component for this part will be a robust learning objective, which will allow us to handle noisy data commonly encountered in unconstrained environments.

Next, we tackled also the problem of concurrent segmentation and superquadric recovery with a Region Proposal Network (RPN) and expanded the entire net into a Faster Region-based CNN. For common layers we used the SqueezeNet architecture which can be learnt much faster than VGG16. The results have shown that with CNNs we can achieve a high accuracy of model parameter reconstruction in nearly real-time.

In the future other CNN architectures for the common layers will be tested, for example ResNet. Eventually, our goal is to develop a CNN solution that is optimal for the planned task. When designing the CNN models we will build on the most recent and successful architectural choices from the literature, such as residual networks, inception networks, attention networks and similar. That means we will develop our own specific architecture, generate appropriate database and use a pretraining paradigm to optimise the outcome.

The issue of the input format for 3D image data should also be researched more closely. Instead of several range images, the input 3D point cloud could be represented with 3D voxels or with a single cylindrical panoramic range image.

ACKNOWLEDGEMENT

Research in this article was partially supported by the Slovenian Research Agency under Research program Computer vision (P2-0214).

REFERENCES

[1] Donatella Dominici, Maria Alicandro, and Vincenzo Massimi. UAV photogrammetry in the post-earthquake scenario: case studies in L'Aquila. *Geomatics, Natural Hazards and Risk*, 8(1):87–103, 2017.

[2] Pengfei Dou, Shishir K Shah, and Ioannis A Kakadiaris. End-to-end 3D face reconstruction with deep neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii*, volume 5, 2017.

[3] Aleksandrs Ecins, Cornelia Fermüller, and Yiannis Aloimonos. Detecting Reflectional Symmetries in 3D Data Through Symmetrical Fitting. In *ICCV Workshops*, pages 1779–1783, 2017.

[4] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust RGB-D object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 681–687. IEEE, 2015.

[5] Žiga Emeršič, Dejan Štepec, Vitomir Štruc, and Peter Peer. Training convolutional neural networks with limited training data for ear recognition in the wild. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 987–994. IEEE, 2017.

[6] Žiga Emeršič, Luka Gabriel, Vitomir Štruc, and Peter Peer. Convolutional encoder-decoder networks for pixel-wise ear detection and segmentation. *IET Biometrics*, pages 1–10, 2018.

[7] Miran Erič, Andrej Gaspari, Katarina Čufar, Franc Solina, and Tomaž Verbič. Zgodnjerimska ladja iz Ljubljanice pri Sinji Gorici = early Roman barge from the Ljubljanica river at Sinja Gorica. *Arheološki vestnik*, 65:187–254, 2014.

[8] A. Garcia-Garcia, F. Gomez-Donoso, J. Garcia-Rodriguez, S. Orts-Escolano, M. Cazorla, and J. Azorin-Lopez. PointNet: A 3D Convolutional Neural Network for real-time object class recognition. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1578–1584, July 2016.

[9] Edward Grant, Pushmeet Kohli, and Marcel van Gerven. Deep disentangled representations for volumetric reconstruction. In *European Conference on Computer Vision*, pages 266–279. Springer, 2016.

[10] Klemen Grm, Vitomir Štruc, Anais Artiges, Matthieu Caron, and Hazım K Ekenel. Strengths and weaknesses of deep learning models for face recognition against image degradations. *IET Biometrics*, 7(1):81–89, 2017.

[11] Riza Alp Güler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. DenseReg: Fully Convolutional Dense Shape Regression In-the-Wild. In *CVPR*, volume 2, page 5, 2017.

[12] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and ¡ 0.5 MB model size. *arXiv preprint arXiv:1602.07360*, 2016.

[13] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3D face reconstruction from a single image via direct volumetric cnn regression. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 1031–1039. IEEE, 2017.

[14] Aleš Jaklič, Miran Erič, Igor Mihajlović, Žiga Stopinšek, and Franc 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 Brač, Croatia. *Journal of Archaeological Science*, 62(10):143–152, 2015.

[15] Aleš Jaklič, Aleš Leonardis, and Franc Solina. *Segmentation and Recovery of Superquadrics*. Computational Imaging and Vision. Springer Netherlands, 2013.

[16] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 253–256. IEEE, 2010.

[17] Aleš Leonardis, Aleš Jaklič, and Franc Solina. Superquadrics for segmenting and modeling range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1289–1295, 1997.

[18] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *CoRR*, abs/1703.09312, 2017.

[19] Blaž Meden, Refik Mallı, Sebastjan Fabijan, Hazım Ekenel, Vitomir Struc, and Peter Peer. Face deidentification with generative deep neural networks. *IET Signal Processing*, 2017.

[20] S Miao, Z J Wang, and R Liao. A CNN Regression Approach for Real-Time 2D/3D Registration. *IEEE Transactions on Medical Imaging*, 35(5):1352–1363, May 2016.

[21] Ricardo Pascoal, Vitor Santos, Cristiano Premebida, and Urbano Nunes. Simultaneous segmentation and superquadrics fitting in laser-range data. *IEEE Transactions on Vehicular Technology*, 64(2):441–452, 2015.

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

[23] Jean Ponce and David Chelberg. Finding the limbs and cusps of generalized cylinders. *International Journal of Computer Vision*, 1(3):195–210, 1988.

[24] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[25] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. In *Computer graphics forum*, volume 26, chapter 2, pages 214–226. Wiley Online Library, 2007.

[26] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2014.

[27] Abhishek Sharma, Oliver Grau, and Mario Fritz. VConv-DAE: Deep Volumetric Shape Learning Without Object Labels. *CoRR*, abs/1604.03755, 2016.

[28] Franc Solina and Ruzena Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):131–147, 1990.

[29] Franc Solina and Ale Leonardis. Proper scale for modeling visual data. *Image and Vision Computing*, 16(2):89 – 98, 1998.

[30] Maciej Stefańczyk and Włodzimierz Kasprzak. Model-Based 3D Object Recognition in RGB-D Image. In Halina Kwaśnicka and Lakhmi C. Jain, editors, *Bridging the Semantic Gap in Image and Video Analysis*, pages 73–96. Springer International Publishing, Cham, 2018.

[31] Žiga Stopinšek and Franc Solina. 3D modeliranje podvodnih posnetkov. In Marko Munih, editor, *SI robotika*, pages 103–114. Slovenska matica, 2017.

[32] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2686–2694, 2015.

[33] George Trigeorgis, Patrick Snape, Iasonas Kokkinos, and Stefanos Zafeiriou. Face normals "in-the-wild" using fully convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 38–47. IEEE, 2017.

[34] Giulia Vezzani, Ugo Pattacini, and Lorenzo Natale. A grasping approach based on superquadric models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1579–1586. IEEE, 2017.

[35] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[36] Allan Zelener and Ioannis Stamos. CNN-based object segmentation in urban lidar with missing points. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 417–425. IEEE, 2016.

[37] Qian-Yi Zhou and Ulrich Neumann. Complete residential urban area reconstruction from dense aerial LiDAR point clouds. *Graphical Models*, 75(3):118–125, 2013.

[38] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3D human pose estimation from monocular video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4966–4975, 2016.