UNIVERSITY OF GENOVA

DITEN - DEPARTMENT OF ELECTRICAL, ELECTRONICS AND TELECOMMUNICATION ENGINEERING AND NAVAL ARCHITECTURE
PAVIS - PATTERN ANALYSIS AND COMPUTER VISION, ISTITUTO ITALIANO DI TECNOLOGIA

PHD IN SCIENCE AND TECHNOLOGY FOR ELECTRONIC AND TELECOMMUNICATION ENGINEERING
CURRICULUM: COMPUTATIONAL VISION, RECOGNITION AND MACHINE LEARNING

Spatial Reasoning for 3D Shape Understanding

PhD Thesis submitted for the degree of *Doctor of Philosophy*
(XXXV cycle)

PhD Candidate: **Mohammad Zohaib**

Alessio Del Bue — Supervisor

Matteo Taiana — Co-Supervisor

Maurizio Valle — Coordinator of the PhD course

DITEN

iit PAVIS
ISTITUTO ITALIANO DI TECNOLOGIA

March 2023

Spatial Reasoning for 3D Shape Understanding

3D Shape Understanding

Mohammad Zohaib

DITEN

University of Genova

This dissertation is submitted for the degree of

Doctor of Philosophy

March 2023

To my family, who always supported me.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified.

Mohammad Zohaib
March 2023

Acknowledgements

Many people deserve acknowledgement due to their support, guidance, and help professionally and personally throughout my PhD degree.

I would like to express my deep gratitude to my primary supervisor, Dr. Alessio Del Bue, for his valuable support, guidance and fruitful discussions throughout my doctoral journey. Under the shadow of his command, I was able to accomplish this milestone.

I am also thankful to Dr. Matteo Taiana, Dr. Milind Gajanan Padalkar and Dr. Pietro Morerio for their help and great suggestions in conducting the research and improving the ideas.

I would like to extend my gratitude to all my colleagues (including Shahnawaz, Shahid, M. Dahy, Abubakar, Usman, Waqar, Valentina, Shafiq, Saber, Davide, Sanket, Giancarlo, Javed, Francesco, Julio, Gianluca, Dario) at PAVIS, who helped me at different levels.

I can not forget the vital support I received from my seniors, especially from Dr. Vaibhav Bansal, Dr. Ahmad Mahmood Tahir and Dr. Jamshed Iqbal. I am thankful to them for their valuable suggestions, advice and encouragement.

In the end, I extend my special appreciation to my family members, who have always been a source of inspiration and gave me strength when I thought of giving up, who continuously provide their moral, spiritual and emotional support.

Mohammad Zohaib
December 2022, Genoa

Abstract

In this thesis, we studied deep learning based approaches to estimate different 3D properties of an object. As a result, we proposed methods that make use of either a single image or a single point cloud to reason about an object's geometry.

We started from a very recent problem, 3D shape reconstruction from a single-view RGB image. We observed that some of the existing methods work for synthetic images only and they fail when they are executed for real images (with background). While other approaches can extract 3D shapes from real images, however, their estimations are not smooth, sharp and complete. By considering the background as a major limitation of the existing methods, we proposed two solutions. The first solution (baseline solution) enables the execution of the synthetic methods for the real dataset. The solution is based on two modules; a segmenter and a reconstruction. The segmenter module takes a real image, segments the object of interest, and pastes the segmented object in the center of the white image. The processed image (which seems similar to the synthetic image) is passed to the reconstructor that estimates the object's 3D shape. We found that the solution has increased the performance of the existing synthetic approaches for real images.

Since the baseline solution is based on a segmenter module, it can not be considered an optimal solution. It is due to the fact that the reconstruction accuracy is totally dependent on the output of the segmenter – if the object is not segmented accurately, the reconstructor will not reconstruct the accurate 3D shape. To solve this problem, we present a second solution that removes the requirement of the segmenter module. Instead of segmenting the object from the image, it separates the features of the object of interest by filtering the features of the background. The object's features are later used to reconstruct the object's 3D shape. The reconstructed shapes are compared with those of the State-Of-The-Art (SOTA) approaches. It is found that the proposed approach outperforms them by estimating comparatively more accurate, smooth, sharp and complete 3D shapes.

The proposed two object reconstruction solutions produce 3D shapes always in the canonical pose. However, for many applications such as object grasping manipulators, pose information is required. Considering that the object pose can be estimated using the keypoints, we conducted research to estimate such keypoints from images in a supervised way and from point clouds in a self-supervised setting.

Our first keypoints estimation approach takes a single-view RGB image as input, extracts pixel-wise features and uses them to estimate keypoints in 3D space. The designed network is trained in a fully supervised way using the ground truth human-annotated keypoints. Moreover, the approach also estimates a confidence score for every keypoint representing its validity. Based on the confidence scores, the network separates valid keypoints from the estimated N keypoints based on the object's geometry. The valid keypoints are used to estimate the relative pose between different views of an object. It is found that the angular distance error of the proposed approach is comparatively lower than that of the SOTA approaches.

The first presented keypoints estimation approach uses only RGB images to estimate 3D keypoints without using any 3D/depth information as input. Thus in some cases, the keypoints are not accurately predicted. Therefore as a second approach, we present a teacher-student architecture to estimate the keypoints from a single-view RGB image. The network is trained in two steps: first, the teacher module is trained to extract 3D features from point clouds, and second, the teacher module teaches the student module to produce 3D features from RGB images that are similar to those achieved from point clouds. During inference, the network only uses only the student module and extracts 2D and 3D features directly from an RGB image to estimate keypoints in 3D space. The keypoints are compared with those of the existing approaches, including the previously proposed keypoints estimation approach. The results show that the keypoints estimated by the proposed approach are more accurate for computing relative pose between different views of an object.

It can be observed that the above two keypoints estimation solutions are fully supervised and require a huge dataset with ground truth human-annotated keypoints. This limits the reusability of the approaches since very limited datasets contain accurate keypoint annotations. Therefore, as a third approach, we present an approach that estimates keypoints in a self-supervised without using any ground truth information. Although estimating keypoints similar to human-annotated ones without supervision is a challenging task, the proposed approach estimates the keypoints that best characterize the object's shape. We achieved this by utilizing a combination of loss components that forces the estimated keypoints towards

the object's surface and prevents them from moving away from the object. The approach is tested for rotated, noisy and decimated point clouds, and it is found that it outperforms the SOTA un-/self-supervised approaches.

Apart from the contributions and comparisons with the competitor approaches, the thesis also presents limitations, possible extensions and real-world applications of the proposed approaches.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Detecting objects in an environment and characterising them with 3D information, such as 3D shape or pose, is an important research area in Computer Vision, with a high impact in multiple sectors including: Augmented/Virtual Reality (AR/VR), medical imaging (CT Scans, MRI, etc.), Autonomous Driving, Human Robot Interaction (HRI) in a collaborative environment (considering sensitive objects in the surroundings, like a human, etc.), biological/chemical science (understanding of cell-to-cell interactions and growth), etc. Estimating the 3D shape or the pose of one object can be done by exploiting different input modalities and can rely on different representations for the objects, depending on the nature of the specific problem that one sets out to solve. We start by discussing these two aspects before going into the details of the problems we addressed in this thesis.

Multiple input modalities, such as RGB cameras, depth cameras, LiDARs, can be considered as input to estimate the shape or pose of an object. The most prominent input modality is the image – with objects being detected and their properties being estimated from RGB images. This is a direct consequence of the fact that cameras are ubiquitous: we have them on phones, on cars, the security cameras are mounted on city infrastructure, in medical instruments like endoscopes, and in industrial setups to guide the robots to perform autonomous tasks such as bin packing and arranging, etc. Depth-enabled cameras are becoming more common, as can be seen in some gaming consoles, as well as some consumer-grade cell phones. They allow such devices not only to capture the 2D image but also the 3D depth information. The depth in such devices can be estimated using either structured light or time-of-flight techniques. The depth is a strong cue for object detection and localisation. LiDAR (Light Detection and Ranging) also captures the distance of the object's surface from the sensor in terms of sparse

points and generates Point Cloud Data (PCD). LiDAR is particularly useful in surveying operations (such as 3D mapping) to generate 3D point clouds of the environment. They are commonly used in robotics (autonomous cars to generate a 3D map of the environment by estimating 3D shapes of the surrounding objects), Astronomy (NASA uses the LiDAR technology to explore space objects), forestry and farming (to monitor the growth rate), augmented reality (to interact with real 3D objects), etc. In some cases, the point cloud of an object is used as input for the 3D pose and shape estimation. Such point clouds can be derived from the combination of multiple input data (RGB, depth images, LiDAR) or from the synthetic models of the objects. In this thesis, we propose methods that use either RGB images or PCDs as inputs for estimating the 3D shape of an object or its 3D keypoints.

An object's 3D shape can be represented in multiple ways, from a 3D bounding box, to voxel grids, to point clouds, to a mesh and to a set of keypoints. The 3D bounding box represents the box that best fits the shape of an object. For example, this representation can be useful in object detection and tracking. However, it may not highlight an object's shape or geometry. On the contrary, the voxel grid represents a shape using 3D cells, similar to the pixels in 2D space. They require huge memory to represent an object (since the memory grows cubically with the resolution), thus, they are discouraged from using in deep learning tasks. In comparison, point cloud based representation is memory and computationally efficient. However, they lack connectivity information between the points and are limited to a fixed number of points. Mesh representation uses vertices and faces to model an object's 3D shape. However, estimating a mesh often requires having a template mesh of the object class [110, 76, 46, 42]. A set of keypoints can be used to represent an object's structure (using the minimum number of possible points). Sets of keypoints are helpful in computing the relative pose between different views of an object or in finding correspondences between different objects. They are easy to handle and process, however, they can not be used to represent the fine details of an object's surface. In recent research, another representation called "implicit function" is used to represent an object. It computes the (continuous) boundary of an object in terms of a non-linear function (such as the Signed Distance Function (SDF)). The working principle is similar to a classifier that classifies an object's inner and outer regions. Generally, meshes are used to visualize shapes estimated/defined by the implicit function.

Depending on the application, it is convenient to use any of the above-mentioned representations. For instance, for a robotic grasping of an object, a rich representation like a mesh could be a better option. While for estimating the relative poses between views of the same

object, a simpler representation like a 3D keypoints can be preferred. In this thesis, we use these two representations to estimate the object's shape and the keypoints that are used to model its structure.

In this thesis, We initially focused on estimating the 3D shape of one object based on one image. This is important, for example, when one wants to model an object and visualize it in virtual reality. The method we proposed improves the accuracy of the shape estimation on real images (as opposed to images with a white background, which are commonly used in the literature) compared to the results achieved by existing methods. Then we shifted our focus to object poses, and for this goal, we decided to use the simpler representation of 3D keypoints. Detecting 3D keypoints for one object given one image is important, especially when only 2D information is available as input, but one wants to reason in 3D space, i.e., computing correspondences between 2D images and the 3D models (2D to 3D matching), aligning 3D shapes with respect to the objects present in images (determining 3D pose, motion capturing and animation for an avatar), generating an object's skeleton, 3D shape deformation, pose transformation for 3D characters, etc. [86, 81, 39, 149, 125, 74, 20, 47, 80, 52, 59]. We proposed a solution that estimates 3D keypoints using only a single image that outperforms the existing approaches by estimating more accurate 3D keypoints. In a continuation of that work, we developed another algorithm to solve the same problem, which exploits the 3D point clouds which are available at training time, following a teacher-student strategy. The comparison with the State-Of-The-Art (SOTA) approaches shows that keypoints estimated by the presented method are comparatively more accurate for relative pose estimation. Finally, we developed a self-supervised approach for estimating 3D keypoints from a point cloud. The importance of this method can be highlighted by the fact that it does not require human-annotated ground truths, which are expensive to acquire.

In the remainder of this chapter, we describe the problems that are addressed in this thesis in detail, present the SOTA approaches that are proposed in the literature to solve the addressed problems, highlight their limitations, and finally present our solutions to the problems. We divide the addressed problems into three sections: 3D shape reconstruction from a single-view RGB image, supervised keypoints estimation from a single-view RGB image, and the self-supervised keypoints estimation from a Point Cloud Data (PCD).

1.1 3D shape reconstruction from a single-view RGB image

Reconstructing 3D shapes from images is considered as a demanding problem in Computer Vision and has actively been tackled by the scientific community. It is due to the fact that the objects' 3D models are required in several real-world applications, including the movie industry, video games, virtual simulated environments, etc. Several deep-learning approaches have been proposed that are based on point clouds, depth, multi-view, or single-view images [13, 116, 146, 45]. Considering the large availability of cameras in our daily life, we aimed to use only a single RGB image to solve the 3D reconstruction problem.

Existing approaches and limitations:

Most of the existing 3D shape reconstruction methods work for synthetic images only and fail to reconstruct objects from real images in the presence of natural background [17, 68, 12]. On the other side, the approaches that reconstruct 3D shapes from real images are not very accurate [31]. Their reconstructions are not smooth, sharp and complete, especially in the presence of occlusions. As an example, the reconstructed shapes of the Mesh R-CNN [31] are shown in Fig. 1.1.

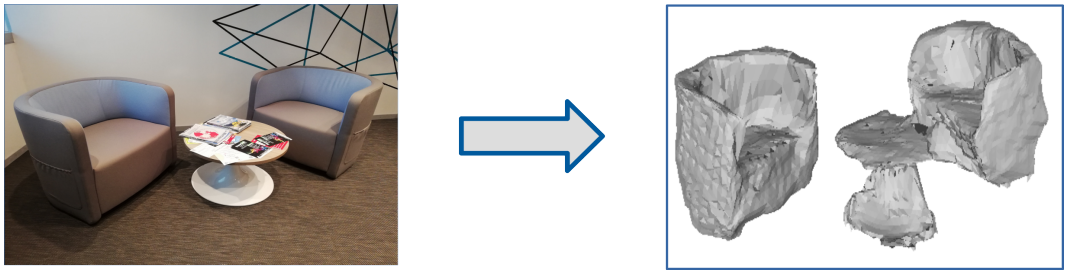


Figure 1.1 Execution of the Mesh R-CNN [31] on a real image containing two mini-sofas and a table. Right, there is an input image where the table is occluding one of the sofas. Left, the output of the approach, the reconstructed 3D shape where the table and the right sofa are not fully reconstructed due to the occlusion. Moreover, the reconstructions are not smooth, sharp and complete.

The image, as illustrated on the left side in the figure, contains three objects: two sofas and a table. The approach separates objects using Mask R-CNN [35] and reconstructs their 3D shape. It can be observed that the right sofa is not reconstructed accurately due to occlusion with the table. Moreover, the reconstructed objects are not smooth and sharp, and the edges/surface are not properly estimated.

Proposed baseline solution:

We first present a baseline solution to enable the existing approaches designed to deal with synthetic images (such as CvxNet [17] and ONet [68]) to reconstruct 3D shapes from real images. This approach is based on two modules: the segmenter and the reconstructor. The segmenter module separates the part of the image that contains the object of interest, applies padding and centering on the separated image part in order to make it similar to a synthetic image – an image with a white background with an object in the center. The reconstructor module uses the processed images (output of the segmenter module) and reconstructs the 3D shape. As a reconstructor, we use the original versions of the CvxNet and ONet. To compare the performance of the CvxNet and ONet with and without the baseline solution, in the first step, we execute them directly on the real images of the Pix3D dataset. In the second step, we integrate them in the baseline solution (in place of the reconstructor module) and test for the same Pix3D dataset. We observed that the baseline solution has significantly improved the performance of the CvxNet and ONet.

Proposed end-to-end solution:

Although the proposed baseline solution can be used to test the existing synthetic approaches on real images, it can not be considered as an optimal solution. That is due to the fact that the solution is not end-to-end since it is based on two different networks that are trained independently. Also, the performance of this solution is completely dependent on the performance of the segmenter, thus it would suffer in the case of inaccurately segmented objects.

To overcome the limitation of the proposed baseline solution, we propose an end-to-end approach that removes the need of a segmenter by computing stable features for an object of interest from a real image by reducing the influence of the image background. Our network achieves this goal by utilising two images simultaneously: a synthetic image with white background and its realistic variant with a natural background. During training, the method uses two encoders simultaneously in order to compute features for the synthetic and the realistic image. Both the features are compared in order to force the model to produce the same features for both input versions. Since the same part of both images is the part presenting the object, the common features represent the features of the object. During the testing, only a single encoder is used to extract the features from the real or realistic image. The extracted features (features of the object) are used to reconstruct the 3D shape, allowing the model to predict an accurate 3D object surface from the image. The

approach is evaluated for both real images of the Pix3D [97] dataset and realistic images rendered from the ShapeNet dataset [9]. The results are compared with SOTA approaches in order to highlight the significance of the proposed approach. Thanks to these contributions, the method we propose achieves a higher accuracy for shape reconstruction than existing methods. This work has been published in the conference ICIAP–2021 [147].

1.2 Supervised keypoints estimation from a single-view RGB image

The method proposed in the previous section estimates the 3D shape of an object, but it does so in a canonical pose, i.e., no information is produced about the pose of the observed object. However, the object's pose plays a vital role in several real-world applications, such as the object grasping with an articulated gripper. Thus we decided to consider the object's pose estimation from a single-view RGB image as the next problem.

Existing approaches and limitations:

In the literature, we found that the pose can be estimated by predicting the 3D keypoints on the object. The keypoints provide an object's structural representation using the minimum number of points, which are easy to process further in comparison to complete 3D point clouds or meshes. Moreover, in some cases, keypoints also contain semantic information by ensuring their unique order. Most of the existing approaches either use point clouds [135, 89, 128, 6] or multiple images (RGB or depth) [98, 28] for computing keypoints. However, we estimate the keypoints using a single RGB image as input.

Proposed solution 1:

In this solution, we estimate an object representation that encodes not only the essence of the shape of one object but also its orientation, i.e., a set of keypoints. In supervised methods, keypoints represent the minimum points that are closer to human annotations. Whereas in un-/self-supervised approaches, they can be considered as the points that best characterize the shape of the object. For example, the minimum possible points that can be considered as the keypoints for the chair category are shown in Fig. 1.2. The last two columns show 12 points selected from the same object, either using Farthest Point Sampling (FPS) [55] (which does not produce keypoints), or selected by a user to represent the shape of the object. It can be observed that the 12 points highlighted in the box do not represent the structure of the

chair. Whereas 12 points shown in the last column best represent the object's shape and thus can be considered as keypoints.

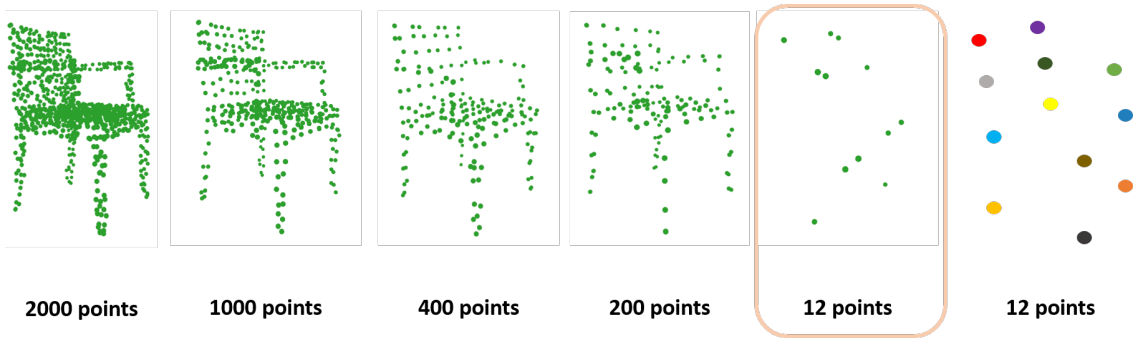


Figure 1.2 Keypoints – the minimum possible points that can represent a geometry of an object. The original object contains 2000 points. The object can be recognized until we down-sample the points up to 200 points using Farthest Point Sampling (FPS) [55]. If we down-sampled the object further, with the same method, the object can not be recognized anymore. In comparison, the last column also contains the 12 points, they can represent the object's structure, and thus can be considered as the keypoints.

In the literature, it is found that most of the existing methods either use point clouds or multiple RGB/depth images to estimate the 3D keypoints. In comparison, we propose an approach which requires only a single-view RGB image. It extracts 2D features from an image and converts the 2D features to pixel-wise representations such that every pixel instead of representing the pixel value represents pixel features. These features are then used to predict 3D coordinates (positions) of the N keypoints. Where N is the total number of predicted keypoints and is predefined. The predicted positions are compared with the ground truth keypoints. To penalize the wrong predictions, we use several loss functions between the predicted and the ground truth keypoints. The network also maintains the semantic order of the keypoints. The order localizes every keypoint with respect to the original object, a property which can be used to establish keypoints correspondences. Moreover, the network also computes a confidence score of every keypoint that enables it to predict a different number of keypoints (out of predicted N keypoints) based on an object's shape. Therefore, unlike existing approaches, the network can be trained to address several categories at once. We compare the relative angular distance error computed between the keypoints sets estimated for the two views of an object by our approach and those of the SOTA approaches. The results show that the error of our approach is lower than that of the SOTA approach. This work has been published in ICIAP–2021 [148].

Proposed solution 2:

The above proposed keypoints estimation method learns from images without considering any 3D information as input. Since, at training time, it is common to have complete point clouds of the object, those can be used to improve the locations of the estimated 3D keypoints. Thus as an extension, we decided to employ a hallucination technique for improving the accuracy of the Keypoints estimates.

In this solution, we designed a knowledge distillation framework that exploits 3D information of the objects during training to improve the keypoints estimation. It is based on a teacher-student network that is trained in two steps: In the first step, the teacher is trained to extract 3D features from a point cloud of an object, which are used in combination with 2D features (of an image of the same object) to estimate the 3D keypoints. In the second step, the teacher teaches the student module to hallucinate the 3D features from the input RGB image that are similar to those extracted from the point cloud. This procedure helps the network during inference to extract 2D and 3D features directly from images without requiring point clouds as input. Similarly to the previous approach, this network estimates an ordered list of keypoints along with their confidence scores. We compare the angular distance error between the keypoints estimated for two views of the same object with those of the SOTA approaches. The results show that the proposed approach remains successful in estimating the keypoints in different poses with minimum angular distance error, with better accuracy compared to the previously described approach. This work is under review in the Pattern Recognition Journal.

1.3 Self-supervised keypoints estimation from a Point Cloud Data

The methods we proposed in the previous sections estimate 3D keypoints of objects from images in a supervised setting. Considering the difficulty in creating the ground truth annotations for this problem, we decided to design an unsupervised method. However, using only images for unsupervised keypoints estimation is a very challenging task, and most of the existing methods use point clouds as inputs. For this part of our work, we choose to do the same, i.e., use PCDs as input.

Existing approaches and limitations:

The literature presents that the keypoints can be estimated in an unsupervised way by taking advantage of the geometric properties of the objects. However, such approaches suffer a

lot in estimating the semantic and aligned keypoints over all the parts of an object's shape and hence their performance reduces in the downstream tasks [89]. Some of the existing unsupervised approaches estimate 3D keypoints and use them to generate an object's skeleton. Although their keypoints well describe the skeleton of the object, they do not characterize its shape [89]. Similarly, some approaches learn to produce keypoints by considering the object's symmetry. Such approaches are very sensitive to the object's shape and fail to estimate good keypoints for asymmetric objects [24]. Considering the above-mentioned limitations, our goal is to estimate semantically consistent keypoints that well characterize the object's shape. They should be independent of the object's geometry, and robust against common perturbations.

Proposed solution:

We propose a new method to infer keypoints from arbitrary object categories in practical scenarios where PCDs are arbitrarily rotated, noisy, and sub-sampled. Our proposed model adheres to the following principles: i) keypoints inference is fully unsupervised (no annotation given), ii) keypoints position error should be low and resilient to PCD perturbations (robustness), iii) keypoints should not change their indexes for the intra-class objects (semantic coherence), iv) keypoints should be close to or proximal to PCD surface (compactness). We achieve these *desiderata* by proposing a new self-supervised training strategy for keypoints estimation that does not assume any a priori knowledge of the object class and a model architecture with coupled auxiliary losses that promotes the desired keypoints properties. We compare the keypoints estimated by the proposed approach with those of the SOTA unsupervised approaches [24, 89]. The experiments show that our approach outperforms them by estimating keypoints with high coverage while being semantically consistent that best characterizes the object's 3D shape for downstream tasks. This is under review in CVPR–2023.

1.4 Contributions

The list of chapter-wise contributions is as follows;

- *3D shape reconstruction from a single-view RGB image*
  We proposed an approach to reconstruct an object's 3D shape in an end-to-end manner from natural images, even in the presence of a background. Unlike the existing approaches, it extracts features from realistic images that are closer to those extracted from similar synthetic images. Our results are comparable to those obtained us-

ing a combination of SOTA methods for segmentation and 3D reconstruction from background-less synthetic images. Moreover, our approach minimizes the requirement of the segmentation approach by separating the object's features from the features of the background.

- *3D keypoints estimation from A single RGB image*
  We proposed an approach to estimate 3D keypoints from a single-view RGB image. Unlike the existing approaches, it estimates a confidence score for every keypoint, which allows the selection of valid keypoints from the set of estimated keypoints. The estimated keypoints also provide order-wise semantic information that is independent of the object's view. It is a flexible approach that can predict geometry based number of keypoints, to accommodate inter-and-intra-class shape variations. Unlike the existing approaches, our approach can be trained for various categories simultaneously and is capable of estimating keypoints of the self-occluded parts of the objects. The estimated keypoints can be used for downstream tasks such as shape alignment and relative pose estimation between two objects.

- *CDHN: Cross-Domain Hallucination Network for 3D keypoints estimation*
  We proposed an approach, as an extension of the method proposed as task 2, to estimate 3D keypoints from single-view RGB images by leveraging information learnt from 3D data during training. The approach presents a way to produce 3D features directly from RGB images without requiring point clouds or depth information. The approach outperforms SOTA approaches for all the selected categories.

- *SC3K: Self-supervised and Coherent 3D Keypoints estimation from rotated, noisy, and decimated PCD*
  We proposed a network to estimate 3D keypoints in the same pose as the pose of the input point cloud, thus minimizes the need of aligned objects. It is robust enough to maintain the correspondences between keypoints estimated for randomly rotated versions of the same object. The keypoints preserve an order-wise semantic consistency between the different objects of the same category regardless of their orientation. The keypoints are estimated close to the object's surface and are well distributed, thus best characterizing the 3D shapes. On average, the presented approach outperforms the SOTA approaches by estimating keypoints on the surface of the objects.

1.5 Publications

1.5.1 Published Papers

- **M. Zohaib**, M. Taiana, M. Gajanan Padalkar, A. Del Bue, "3D keypoints Estimation From Single-view RGB Images", 21st International Conference on Image Analysis and Processing, Italy, pp. 27 – 38, 2022.
  DOI: https://doi.org/10.1007/978-3-031-06430-2_3
- **M. Zohaib**, M. Taiana, A. Del Bue, "Towards Reconstruction of 3D Shapes in a Realistic Environment", 21st International Conference on Image Analysis and Processing, Italy, pp. 3 – 14, 2022
  DOI: https://doi.org/10.1007/978-3-031-06430-2_1

1.5.2 Under-review Papers

- **M. Zohaib**, A. Del Bue, "U3DK: Unsupervised 3D Keypoints Estimation from Rotated, Noisy, and Decimated Point Cloud Data", IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023
- **M. Zohaib**, M. G. Padalkar, P. Morerio, M. Taiana, A. Del Bue, "CDHN: Cross-Domain Hallucination Network For 3D keypoints Estimation", Pattern Recognition journal, 2022

1.6 Thesis organization

The thesis is organized as follows.

Chapter 2 reports the literature on the existing approaches. We divide the chapter into three sections. First, we discuss approaches related to the 3D reconstruction task; second, we present supervised and un-/self-supervised approaches for 3D keypoints estimations from images and PCDs; third, we provide details of the datasets used in this research along with our modifications.

Chapter 3 presents the first task, 3D shape reconstruction from a single RGB image. It highlights the problems associated with the 3D reconstruction approaches and presents two solutions; the baseline solution to execute existing approaches and the end-to-end solution

that extracts features of the object in the presence of the background to estimate accurate 3D shape.

Chapter 4 reports the second task, the supervised approach for 3D keypoints estimation from RGB images. It describes the methods proposed in the literature that either use point clouds or multiple modalities to estimate 3D keypoints and then shows how the proposed approach estimates the same keypoints from only RGB images. The presented approach achieves this goal by combining different loss functions to localize the keypoints at a specific position with respect to the ground truth annotations.

Chapter 5 describes the third task, CDHN: Cross-Domain Hallucination Network For 3D keypoints Estimation, as an extension of task 2. It presents the two steps of the training procedure, teacher and student training, and highlights the process of knowledge distillation from the teacher to student module. This training process allows the network to infer 2D and 3D features directly from RGB images to estimate more accurate 3D positions of the keypoints.

Chapter 6 presents the fourth task, SC3K: Self-supervised and Coherent 3D Keypoints estimation from rotated, noisy, and decimated PCD. It highlights the limitations of the existing supervised approaches, i.e. the requirement of huge datasets containing human-annotated 3D keypoints, which are difficult to generate and can consume a lot of resources. As a solution, it then reports the un-/self-supervised approaches that estimate the 3D keypoints without using the ground truth keypoints. Finally, by describing the constraints of the available solutions, it presents a self-supervised approach that estimates coherent 3D keypoints from rotated, decimated and noisy PCDs.

Chapter 7 concludes the thesis by providing a short description of every chapter. It reports the problem tackled in the conducted research, highlights their limitations and presents the possible future directions.

Chapter 2

Literature Review

This chapter presents the most recent works that are closely related to the two followed research directions; 3D reconstructions and 3D keypoints estimations. Moreover, the datasets used in the experiments are also described.

2.1 3D shape reconstruction from an image

Estimating the 3D shape of an object from images is considered a challenging task that still needs to be fully explored. Part of the interest is generated by the fact that emerging technologies such as AR and VR are demanding high-quality reconstruction results. Considering the wide range of applications, the scientific community has been focusing on reconstructing the object's 3D shapes using geometrical reasoning. As a result, several approaches have been proposed in the last few years.

Mescheder *et al.* [68] presented ONet, a system for 3D reconstruction based on a continuous 3D occupancy function. It discretizes a volumetric space continuously by evaluating occupancy probability. Deng *et al.* presented CvxNet [17], an approach for geometry representation based on primitive decomposition. It models the shape of one object as the union of a small set of convex components. The number of convexes needs to be specified at training time, which negatively affects the flexibility of the reconstruction system. BSP-Net [12], a similar approach proposed by Chen *et al.*, generates 3D meshes via binary space partitioning, relieving the need for specifying the number of components. It performs a recursive subdivision of space to obtain convex sets, thus not relying on a fixed number of elements for object decomposition.

The approaches mentioned above focus on solving the 3D object reconstruction problem in a simplified setting: the input images are obtained by projecting 3D object models onto images with a white background. In an effort towards extending the capabilities of reconstruction algorithms to natural images, Wu *et al.* [114] substituted the white background in their input data with randomized real images. A decisive step towards object reconstruction in the wild was taken with the disclosure of the Pix3D benchmark [97], a dataset composed of natural images for which the poses of the visible objects have been accurately estimated, exploiting a combination of human labelling and automatic optimization. An approach ShapeHD [115] combines deep volumetric convolutions networks with adversarially learned shapes before estimating the 3D shapes. This approach is similar to [114], however, it uses an additional module "deep naturalness model" that penalizes the shape estimator if the estimated shape is unnatural. Pix2Vox++ [120] exploits multiple views of one object to generate multiple 3D shapes. It reports results on synthetic images and natural images from Pix3D, restricted to the 'chair' category. 3D-GMNet [122] reconstructs 3D shapes from a single image using Gaussian distribution. The technique reduces memory footprint compared to those use volume-based occupancy estimation. PGNet [137] presents an approach that takes a single RGB image and semantic projections of an object's parts (partial projection images) as input and reconstructs the object's parts separately using a recurrent generative network. In the end, all the reconstructed parts are combined to gather in order to form a complete object's shape. Other similar works [29, 23, 75, 124] also learn to produce object 3D shapes in the presence of background. However, they aim to estimate shapes in the form of either a point cloud or a voxel grid. Both representations have their own limitations; point clouds lack connectivity information while memory requirements grow cubically with resolution in voxel grids.

On the other hand, SIST [44], a self-supervised approach for an image to 3D shape translation, uses an implicit field decoder to estimate a continuous object's 3D surface. However, SIST was not tested on natural images but on Pix3D images whose background had been painted white, exploiting ground truth segmentation information. Mesh R-CNN [31] separates itself from the rest of the field because it was designed to work on natural images. As impressive as its performance is, Mesh R-CNN achieves a lower reconstruction accuracy with respect to the other existing approaches, e.g., ONet and CvxNet. Salvi *et al.* presented a method that improves ONet by introducing a self-attention module in the encoder [85]. Whereas the decoder is the same as ONet. Although they have tested the approach on natural images taken from the online product dataset [71], they have not presented any quantitative results. Secondly, it is obvious since ONet is trained on synthetic data, it performs poorly on images with a background at inference.

2.2 3D Keypoints estimation

3D keypoints have been used in several geometrical problems as they require minimum processing and are easy to handle in comparison with complete point clouds or meshes. [88]. Moreover, they highlight the most important regions of an object's 3D structure, i.e. corners, joints, etc. [2, 109, 139]. Furthermore, in some cases, they also contain semantic information by ensuring their unique order. In the literature, different modalities have been used for 3D keypoints estimation in 3D space, including point clouds, depth and RGB images. Based on these modalities, some of the recent approaches are described below.

2.2.1 Image-based 3D keypoints estimation

Considering only the RGB images increases the complexity of the 3D keypoints estimation task. Such approaches are less accurate compared to methods that use depth images or 3D point clouds [8]. KP-Net [98], proposed by Suwajanakorn et al., trains models for each object category by computing 3D keypoints from dual view images. During inference, it uses single-view RGB images. However, their estimates are in the form of 2D pixels and depths. The authors present results for four different versions of their approach: 1) supervised KP-Net that learns from ground truth 2D pixels and corresponding depths, 2) supervised KP-Net with a pretrained Orientation Network (O-Net) that provides an object's orientation information, 3) KP-Net (unsupervised) with O-Net, and 4) KP-Net without O-Net. Their results show that the unsupervised KP-Net without O-Net estimates the most accurate keypoints. The method in [148] estimates keypoints from RGB images directly in a 3D space. Although it utilizes the original point clouds for computing the loss functions, it does not utilize them for feature extraction. Thus due to the lack of 3D features, some keypoints are not estimated at the appropriate 3D positions.

2.2.2 Using 2D keypoints for reasoning in 3D space

Some of the recent works have used 2D keypoints estimation for reasoning on an object's 3D geometry as a further inference step. An approach that uses 2D keypoints for 3D object detection is presented in [3]. The approach estimates 2D keypoints from an RGB image and transforms these keypoints into a 3D model of an object taken from a predefined set of 5 CAD templates only. The intrinsic camera parameters are known. Lu et al. present an approach that uses keypoints for finding the pose of a robotic arm [64, 65]. Initially, keypoints are sampled on the kinematic chain and are filtered in order to select optimal

ones using RANSAC [25]. A similar approach that finds semantic correspondence between two images using both appearance and geometry reasoning by incorporating 2D keypoints is presented by Han et al. in [34]. The approach uses these semantic correspondences to produce a warped version of two images. Another approach, presented in [104], estimates semantic 2D keypoints for visual representation. In supervised training, the approach uses 2D ground truth labels. Given a camera calibration, the approach estimates 3D positions and projects them to a 2D plane for computing loss. However, in self-supervised training, it uses multiple images of a scene simultaneously. A similar 3D object detection approach, "SMOKE", is presented in [61]. It defines a 2D keypoint on the image plane that represents center of the 3D object. For 3D to 2D projection, known camera parameters are used. Zhou et al. present the "StarMap" [145] approach that estimates 3D keypoints, which depend on the 2D keypoints/heatmaps. The approach is evaluated for the subset of the validation set that is non-truncated and non-occluded objects. Moreover, the presented results validate that the approach does not consider the occluded areas of an object, e.g., the hidden parts (i.e., the back side of a car or the back legs of a sofa).

2.2.3 RGBD-based keypoints estimation

Some works use multiple modalities including RGB and depth images (RGBD). The point-wise 3D keypoints voting network (PVN3D) [37] estimates 3D keypoints by fusing the appearance and geometrical features extracted from RGBD images. The estimated keypoints are used to compute an object's pose in six Degrees of Freedom (6-DoF) by applying the Least Squares fitting algorithm. Georgakis et al. [28] present an approach that uses RGB and depth images to compute object 3D pose by matching predicted keypoints to the corresponding CAD model. Another RGBD image based approach is presented in [107] that uses estimated 3D keypoints for tracking an object's pose. Although their network does not require 3D shapes during training, during inference the method can work only with objects that are relatively similar to those used in training [19].

2.2.4 Point-cloud-based keypoints estimation

In literature, most of the approaches use point clouds for estimating 3D keypoints. Several approaches have been proposed to estimate 3D keypoints in a supervised way [113, 37, 136, 56, 148, 53, 130]. Such approaches require human-annotated datasets, which are limited to a smaller set of objects since annotating shapes is a time-consuming process. Therefore, the research community is focusing on estimating the same 3D keypoints similar to human

annotations in un-/self-supervised ways, without using the ground truth labels. Some of the recent approaches are presented here.

Liu et al. present an approach [57] that uses 3D keypoints features extracted directly from point clouds to solve object recognition and 6-DoF pose estimation. Their framework consists of two phases. In the first phase, it generates a database of the keypoints sampled from synthetic point clouds. In the second phase, the features extracted from the test scenes are matched to the database using the K-D tree voting method for object recognition. Shah et al. use the keypoints for surface representation in [87]. Their approach estimates 3D keypoints and computes the geometrical relationships between them by considering their relative distance. Based on the minimum distance, subsets of the keypoints are selected that are used in surface representation. Chen et al. present an approach that learns to identify semantically consistent points in the same category in an unsupervised way from an object's PCD [11]. Their network is based on the PointNet++ backbone [79] that assigns a probability (of being a keypoint) to each element of the PCD. The final keypoints are computed using a convex combination of the points weighted by the probabilities. Yuan et al. present an approach that uses two different objects of the same category to estimate semantically ordered 3D keypoints [131]. Jakab et al. [41] use 3D keypoints (unlike [138] that uses the complete point cloud) for aligning two shapes. Their network takes two shapes and finds the keypoints for shape deformation from a set of randomly sampled surface points.

Chen et al. [70] present an unsupervised approach that computes keypoints from the object's point cloud to represent good abstraction and approximation of the input 3D shape. Li et al. present an approach that first generates another variant of the PCD by random transformation and then utilizes both PCDs for estimating the keypoints. Their network first generates clusters from the input point clouds and then it estimates a keypoint for every cluster [48]. A similar approach is presented by Sun et al. in [95]. During training, their network takes two randomly rotated versions of a PCD and computes K capsules containing the attention mask for every point in the input PCD and the corresponding features. Based on the attention masks, points are arranged to K parts of the object. Fernandez et al. present an approach that estimates symmetric 3D keypoints from PCD [24]. The network estimates N nodes (keypoints) and applies non max-suppression for selecting the final keypoints. However, the approach is very sensitive to object symmetry. Also, its performance may decrease for irregular shapes, i.e. airplanes or guitars, whose geometries vary consistently within the category [89]. The authors in [89] present "Skeleton Merger" (SM) to detect aligned and semantic keypoints from PCDs in an unsupervised fashion. It uses the keypoints to generate

a skeleton of the object. Both keypoints and the skeleton are used to reconstruct the PCD. A similar approach, LAKe-Net [99], uses the keypoints for the shape completion task. It localizes the aligned keypoints (using an unsupervised detector), generates surface-skeleton using the keypoints, and uses them to refine the object's shape. SK-Net, proposed in [117], generates random spatial keypoints in 3D space and converges them to an object's point cloud by learning geometric features. Unlike the other approaches, the spatial keypoints are not a part of the object's point cloud. Another similar approach that finds correspondences between different objects of the same category is presented in [14]. You et al. [127] present a method that uses geodesic consistency loss for producing dense semantic embeddings among different objects of the same category. They also create a new dataset "CorresPondenceNet" (CPNet), by annotating the keypoints that semantically correspond between the intra-class objects.

2.3 Datasets

We divide the datasets into two parts considering the different natures of the conducted research. First, we describe the datasets that are used in the 3D reconstruction task. Such datasets commonly contain images and corresponding objects' 3D models. Second, we present datasets that are used in keypoints estimations. Such datasets contain ground truth labels in the form of human-annotated 3D keypoints. We also present our modifications in the existing datasets in order to evaluate the proposed approaches. The details of the datasets are as follows.

2.3.1 3D Reconstruction

In the 3D reconstruction task, we mainly used three datasets. We used the original synthetic ShapeNetCore.v1 [9] to train and test the existing synthetic approaches such as [68, 17]. However, the approach we proposed requires realistic images (synthetic images with real backgrounds). So to generate a new realistic dataset, we used synthetic images from ShapeNetCore.v1 and background images from the SUN [119]. Whereas to evaluate our approach or real images, we use the Pix3D [97] dataset. However, for some categories (i.e. car, airplane, bench, etc.) images are taken from PASCAL [118] and COCO dataset [54].

ShapeNet: ShapeNet is a widespread dataset that is commonly used for 3D reconstruction problems. It contains 3D CAD models along with textures organized in WordNet hierarchy. We use a subset of the dataset provided by Choy *et al.* [16] for training our model. The

subset contains models for 13 categories, with images rendered from 24 random viewpoints, and occupancy points/labels.

SUN2012: The SUN [119] dataset is a collection of indoor, outdoor, Urban, and Nature scenes. The dataset is widely used for scene understanding, object detection, classification, etc. We use a subset of the dataset "SUN2012" that contains 16,873 images. We randomly select images from the pool and use them as a background for generating realistic images.

Pix3D: We selected Pix3D [97] as the benchmark for evaluating the proposed approach because it contains natural images with accurate pixel-level aligned 3D models. Pix3D contains 395 3D models of nine different categories. Tab. 2.1 shows the number of images and models per category in the Pix3D dataset.

Table 2.1 Number of images and models per category in the Pix3D dataset.

Category	Bed	Bookcase	Chair	Desk	Misc	Sofa	Table	Tool	Wardrobe	Total
Images	994	361	3839	700	68	1947	1870	47	143	9969
Models	20	17	221	23	13	20	63	8	10	395

The data in the Pix3D dataset is not homogeneous: part of the images are natural, while others depict the foreground object on a white background.

2.3.2 Keypoints estimation

For the supervised and unsupervised 3D keypoints tasks, we use the KeypointNet [130] dataset. In unsupervised settings, the labels are not used. The details of the dataset and the changes we made to train the proposed networks are given below.

KeypointNet: The keypointNet dataset contains 8329 3D models of 16 object categories, corresponding point clouds, and 83231 keypoints. All the objects in the dataset are in canonical form, always at the origin and in the canonical pose. Moreover, the dataset does not contain images similar to those present in the ShapeNet subset rendered by Choy *et al.* [16].

By considering the procedure followed by Choy *et al.* [16], we render images in 24 views by placing the object's models (of the KeypointNet dataset) at the center of the reference frame and cameras at 24 different locations pointed towards the origin. However, the camera locations are not the same (fixed) for all objects, i.e. locations are randomly selected for every object. Tab. 2.2 shows the number of cameras used in the image rendering for the airplane

category. It can be observed that 98.54% and 99.01% of the test and validation cameras are at positions not seen during training, respectively. For better understanding, the positions of the

Table 2.2 Camera positions for the airplane category. The unique positions are computed with respect to the positions of the cameras in the training set. More than 90% of the test and validation cameras positions are not seen during the training.

	Objects	Cameras	Unique positions
Train	715	17160	–
Test	205	4920	4848
Validation	102	2448	2424
Total	1022	24528	7272

cameras in the 3D space are shown in Fig. 2.1. The green, red, and purple dots illustrate the camera positions for the train, test, and validation set, respectively. Only a fraction (30%) of the unique camera locations are displayed in the figure for the sake of a better visualization.

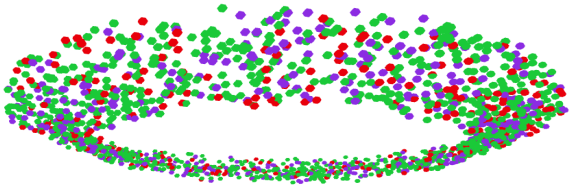


Figure 2.1 Positions of the cameras in 3D space. The green, red, and purple dots illustrate the camera positions for the train, test, and validation set, respectively. Only a fraction (30%) of the unique camera locations are displayed for the sake of a better visualization.

Moreover, for data consistency, the object's point clouds and the ground truth keypoints are also required to transform in the same poses that are used to render the images. This allows us to train the network by feeding images, point clouds and ground truth keypoints in the same pose to the network, such that it can estimate keypoints in different poses. Therefore, we use the same camera parameters (used for the images rendering) and transform the original point clouds/keypoints in the same 24 poses. The same data split as provided by KeypointNet is used in all the experiments.

Chapter 3

3D Shape Reconstruction From A Single-view RGB Image

This chapter presents the 3D reconstruction task from a single RGB image. It highlights the significance of the task in real applications, reports limitations of the existing approaches and proposes an approach to solve the addressed limitations. Estimating 3D object shapes from images is a relevant problem that has actively been tackled by the scientific community. As a result, several approaches have been proposed in the last few years. Most of those approaches are valid for synthetic images with no background or with white background. Such approaches fail to estimate 3D object's shape for real images containing the natural background. Considering the background as a vital limitation for the reconstruction approaches, we aimed to propose an approach that can tackle the background. To make it easy to understand, in this chapter, we differentiate color background images as; real and realistic. Where a real image represents a natural image, i.e., an image with real background and a real object. Realistic image refers to that generated synthetically by rendering the synthetic object in front of the real background, i.e., the image with a real background containing a synthetic object at the center.

Recent advancement in computer vision and deep learning has brought 3D object reconstruction to a level useful for a variety of applications, including Augmented Reality (AR), autonomous driving, robotics applications and game development. Reconstruction approaches in the literature rely on data from one or more input modalities: from simple images [112, 68, 17, 31, 94], depth images [123, 33], to point clouds [32, 126, 62]. The field of 3D object reconstruction from 2D images is evolving quickly, with part of the field

focused on estimating very accurate 3D shapes in simplified settings, i.e., using images with a blank background as input, while another part strives to estimate 3D object shapes in the wild.

Some of the most recent efforts in 3D object reconstruction from images have been focused on solving a simplified problem: reconstructing objects from synthetic images which present no background clutter and perfect foreground/ background segmentation [68, 17]. Data for training and evaluating such systems is collected by rendering 3D object models on a white background. A commonly used source for the 3D models is the ShapeNet dataset [9]. Object reconstruction systems trained on this kind of data achieve high reconstruction accuracy, however their performance drops dramatically for natural images. On the other hand, some of the current research approaches focus on reconstructing 3D objects from natural images [31]. These systems have the advantage of being applicable to real-world images, but this comes at the cost of lower reconstruction accuracy.

In comparison in this chapter, we propose an end-to-end approach that estimates the 3D shape of an object from a real image. The approach learns to generate features from an image with a background that are similar to the features that would be generated from the same image without a background. These features allow the model to estimate a comparatively accurate object shape from a real image. During training, both versions of the image are fed to an encoder in parallel that extracts common features. Whereas during inference, only the real images are used. A sketch of the proposed approach is illustrated in Fig. 3.1. The main contributions are as follows:

- The proposed approach reconstructs 3D shapes in an end-to-end manner from natural images, even in the presence of a background.
- Unlike the existing approaches, our approach extracts features from realistic images that are closer to those extracted from similar synthetic images without background.
- Our results are comparable to those obtained using a combination of State-Of-The-Art (SOTA) methods for segmentation and 3D reconstruction from background-less synthetic images.
- The presented approach does not require any segmentation approach.

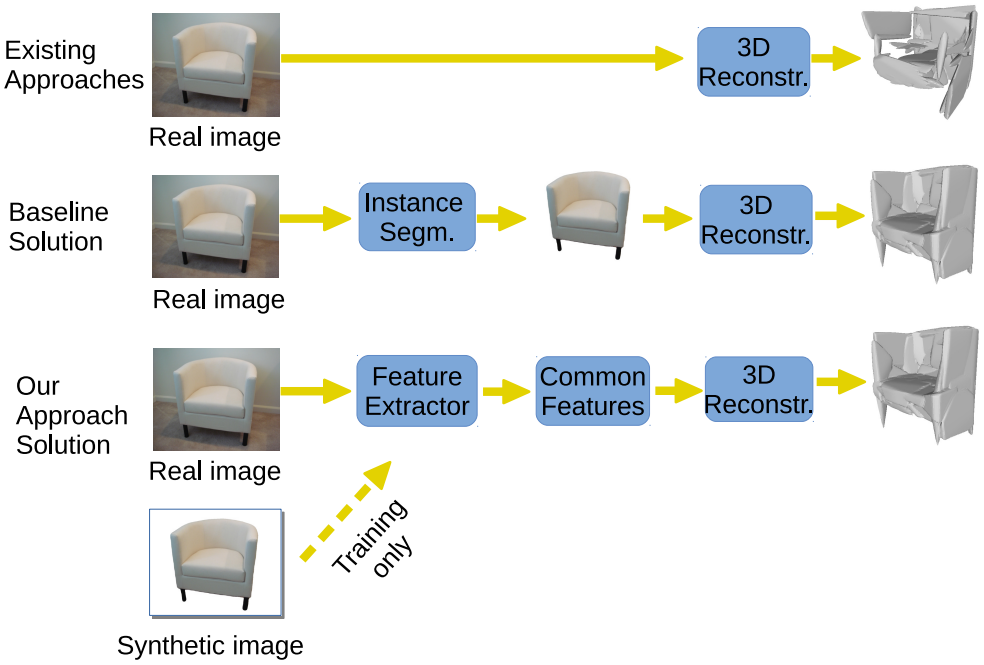


Figure 3.1 Overview of the presented approach. Reconstruction systems with high accuracy do not perform well when applied to natural images directly (top). An instance segmentation algorithm can be considered as a simple solution for removing the background (middle). In comparison, the proposed approach reconstructs accurate 3D shapes by estimating common features for realistic and white background images (bottom).

3.1 Methodology

Given a single real image, the research aims to estimate an accurate 3D shape for the depicted object. In this regard, an end-to-end approach is proposed that learns to extract object features from realistic images – containing a synthetic object with a background. These features allow the approach to decode comparatively accurate 3D shapes from natural images. The designed model is inspired from Occupancy Network (ONet) [68]. It is based on two main modules: feature extractor and 3D shape predictor. The first module extracts object features in the presence of a background, whereas the second module estimates a 3D object boundary in the form of an occupancy function. The process is described in detail in the following subsections. The architecture of the proposed method is illustrated in Fig. 3.2.

Feature extractor

The module extracts object features that are required for 3D shape reconstruction. It is based on an encoder that takes two images, a synthetic image along with its variant with added background, in parallel fashion and produces two feature vectors. The vectors are

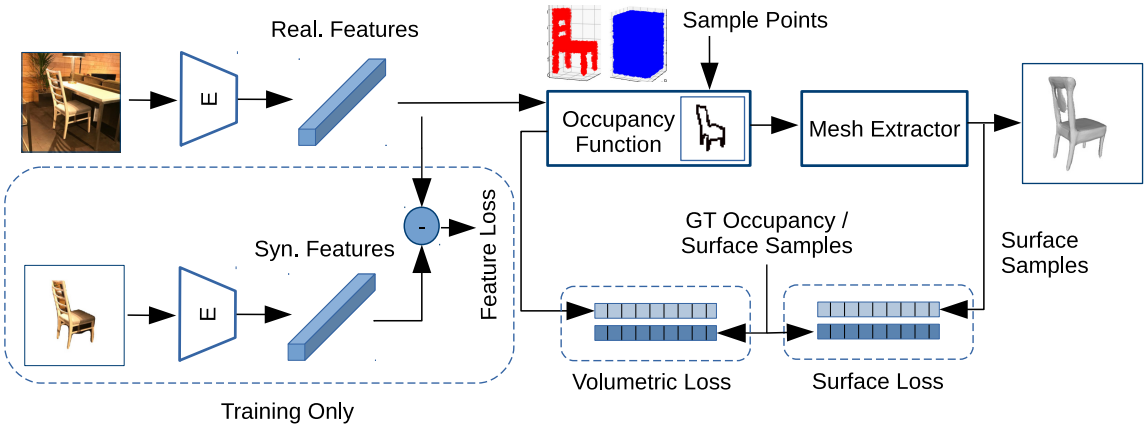


Figure 3.2 Architecture of the proposed approach. During training, an image with and without a background is fed to the encoder in parallel, producing two feature vectors; synthetic and realistic. The vectors are compared in order to enable the encoder to extract common features from both image versions. The shape predictor module based on an occupancy function uses the feature vector (coming from a realistic image) for object boundary estimation. The boundary is evaluated by computing volumetric and surface loss. At inference time, only the real image is fed to the system.

compared in order to instruct the encoder to produce common features for both images. In the beginning, the encoder produces different features; however, after some learning iterations, it starts extracting the same features. We calculate Mean Absolute Error (MAE) for features comparison as

$$\mathscr{L}_{enc} = \frac{1}{M}\sum_{m=1}^{M} |f_m^{(S)} - f_m^{(R)}|, \tag{3.1}$$

where $f_m^{(S)}$ and $f_m^{(R)}$ denote features extracted from synthetic and realistic image. m are the indices of the elements of a feature vector of size M. The loss $\mathscr{L}_{enc}$ is only used to update the weights of the encoder. The process of $\mathscr{L}_{enc}$ computation and the weights update is highlighted in the left part of Fig. 3.2.

Shape predictor

The shape predictor estimates an object's 3D shape by utilizing an occupancy function in a similar way as defined in ONet [68, 85]. The function $\mathscr{F} : R^3 \rightarrow [0;1]$ evaluates N uniform sample points in space for estimating occupancy probability for each point. Where 0 and 1 represent if the samples are outside or inside the object boundary, respectively. It is a combination of 5 fully-connected ResNet blocks. Each block contains a pair of Conditional Batch-Normalization (CBN), ReLU activation functions, and a fully connected layer. The output of the last block is downsampled to a 1D vector and passed through a sigmoid

activation function in order to obtain estimated probabilities. For a detailed visualization of the occupancy function see Fig. 2 of [85]. The shape predictor takes common features extracted by the feature extractor and a set of uniformly sampled 3D points. It first computes 256D features for every point using a fully connected network and passes them along with common features to the occupancy function. The function estimates volumetric occupancy for every point with respect to the object's boundary. The estimated occupancies indicate whether the points belong to the object or its surroundings. These occupancies are compared with corresponding GT values. We use Binary Cross-Entropy (BCE) Loss for comparison as

$$\begin{aligned} \mathscr{O} &= \mathscr{F}(f_{com}, P_i) \mid i : 1 \; to \; N \\ \mathscr{L}_{vol} &= \mathscr{L}_{BCE}(\mathscr{O}, \mathscr{Q}), \end{aligned} \tag{3.2}$$

where f_{com} represents common features, P_i are the N 3D points for evaluation and the corresponding output of occupancy function for all the points is depicted by vector $\mathscr{O}$. The volumetric loss $\mathscr{L}_{vol}$ is a BCE loss between predicted ($\mathscr{O}$) and label ($\mathscr{Q}$) occupancies. The loss improves the geometry of the predicted shape by reasoning on the 3D volumetric space. The mesh extractor module produces meshes in a two-step process; by applying Multiresolution IsoSurface Extraction (MISE) [68] that utilizes occupancy function in order to achieve the required resolution and by using the Marching Cubes algorithm [63] for final mesh extraction.

Additionally, in order to improve the surface of the predicted shape, a surface loss is introduced. Since comparing meshes is a complex and resource consuming task, random points are sampled for predicted (X) and GT (Y) mesh. These sample points are used to calculate Chamfer L_1 Distance as

$$\mathscr{L}_{surf} = \frac{1}{n_X} \sum_{x \in X} \min_{y \in Y} |x - y| + \frac{1}{n_Y} \sum_{y \in Y} \min_{x \in X} |y - x|, \tag{3.3}$$

where x and y represent a single sample from set X and Y, respectively. n_X and n_Y show the total number of samples in each set, which is the same in our case. The overall shape prediction loss can be defined as

$$\mathscr{L}_{overall} = \mathscr{L}_{vol} + \mathscr{L}_{surf}. \tag{3.4}$$

Unlike encoder loss, the shape prediction loss ($\mathscr{L}_{overall}$) contributes in updating the weights of the whole model.

Inference

At inference, the lower part of the feature extractor is removed from the model as highlighted in Fig. 3.2. Real images are fed to the encoder for 2D feature extraction. The extracted features are then used by the second module for producing occupancy probabilities for all the sample points in a 3D volumetric space.

3.2 Experimental setup

In this section, we describe the experimental details we used to assess the performance of the proposed approach in comparison with SOTA. The proposed approach is validated using two types of data; real from Pix3D and rendered from ShapeNet in the presence of a background.

3.2.1 Implementation details

Unlike [85], in our approach, we use ResNet18 [36] encoder (without self-attention module) with pre-trained weights for ImageNet dataset [18]. The last layers are modified to obtain a 256D feature vector. We train the model using synthetic and realistic images that are rendered from ShapeNet objects on white and random backgrounds, respectively. We set the learning rate and weight decay to 1e-5. The rest of the hyperparameters are set to the same values as used in [68].

3.2.2 Datasets

Mainly three datasets are used in the experiments. The details of the datasets can be found in Sec. 2.3 of Chap. 2. For training the proposed network, we use synthetic images rendered by Choy *et al.* [16] and realistic images that are generated by adding random backgrounds to the synthetic images. The background images are taken from the SUN dataset [119]. For evaluation on real images, we use the Pix3D dataset [97]. We select only the object categories in Pix3D for which ONet, Mesh R-CNN, and YOLACT [7] were trained, i.e. tables, chairs, and sofas. The details of the selected test samples from the Pix3D dataset based on a comparison of YOLACT and Mask R-CNN [35] segmentation are provided in Sec. 3.4. For evaluation on some other categories (e.g., car, airplane, bench), real images are taken from PASCAL [21] and COCO dataset [54].

3.2.3 Performance measurement

We use the Chamfer L_1 distance and F_1 score for evaluation. The Chamfer distance computes the average distance between predicted and ground truth meshes with the help of surface sample points. We select 100k sample points randomly from the surface of the predicted and the ground truth mesh and use a KD-tree to associate each point with its nearest neighbour from the other mesh. The final value for the Chamfer distance (as depicted in Eq. 3.3) is an average of all the absolute L_1 distances measured in both directions: from the estimate to the ground truth and vice versa. The F_1 score describes how accurate a prediction is, given the tolerance factor τ (surface thickness) and is computed as the harmonic mean between precision and recall. Precision is defined as the fraction of points that have a distance below τ when going from the estimated mesh to the ground truth mesh. The recall is defined as the fraction of points that have a distance below τ when going in the opposite direction. We chose the value of 0.001 for τ from the ones present in the literature because it is the one that better highlights the difference in performance in our experiments. Better performances correspond to lower values of the Chamfer L_1 distance and higher values of the F_1 score. We report F_1 score values as percentages of its range $[0, 1]$.

3.2.4 Performance evaluation criteria

Comparing the performance of the proposed system with that of Mesh R-CNN [31] is not trivial: CvxNet [17] and ONet [68] are trained to produce estimates in a canonical pose and of a specific size, while Mesh R-CNN produces estimates whose pose is dependant on the viewpoint of the input image. Furthermore, the size of the reconstructed objects is not comparable in the two cases. To ensure a fair comparison, we define an evaluation criterion that first normalizes the ground truth object models and the estimated shapes so that the longest side of any shape has a length 1. To counter the effect of pose variability in the estimates from Mesh R-CNN, we applied a registration algorithm (Iterative Closest Point, ICP) to align each estimate with the ground truth model. Now, since all the shapes are in the same size and pose, they can be used for performance evaluation.

3.2.5 Baselines

We compare our results with SOTA baselines including Mesh R-CNN, CvxNet, and ONet. We use a pre-trained model of Mesh R-CNN. Whereas, CvxNet and ONet are trained on the realistic dataset. This is done because the model trained on white background images

could never perform well on natural images. Additionally, we also present a setup that enables CvxNet and ONet, which are trained on the synthetic dataset, to produce 3D shapes in the presence of a background. The setup integrates an instance segmenter before the 3D reconstruction module. The segmenter processes natural images, partitioning the pixels into two groups: foreground and background. The foreground pixels represent segmented objects. We separate an object of interest and use the segmentation information in order to paint the background pixels with a uniform white color. In order to make it identical to a synthetic sample, we center the object and apply padding by considering the dimensions of the images used during training. The resulting image is then processed by the reconstruction module, which outputs an estimate of the 3D shape of the object. The proposed system is presented graphically in Fig. 3.3.

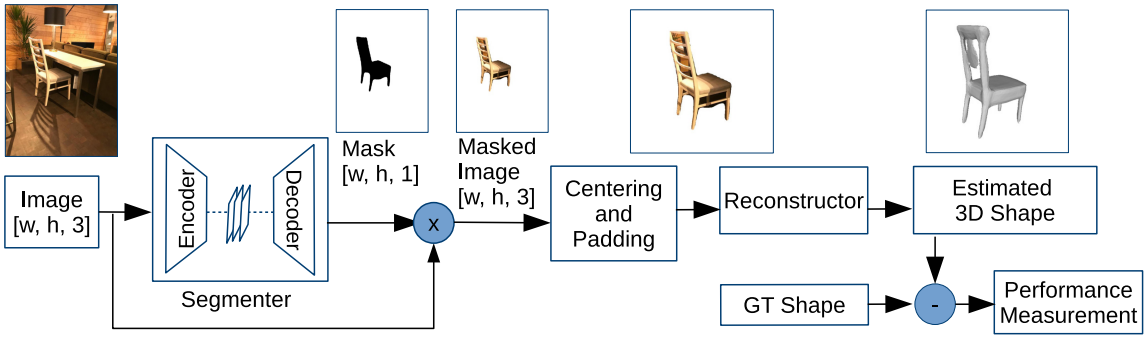


Figure 3.3 Setup to execute CvxNet and ONet on real images. The approaches can produce good results for real images if the input image is processed appropriately; separating an object by applying an instance segmentation algorithm, pasting it on the center of the white image, and padding the image in order to make it similar to synthetic.

We chose YOLACT [7] as the instance segmentation algorithm for the system. For differentiating from the original versions CvxNet and ONet that are trained on the realistic dataset, we are using the letter M (mask) with them. So, the CvxNet-M and ONet-M show the versions when the segmenter is integrated with them.

To evaluate the performance of the CvxNet-M and ONet-M with respect to the CvxNet and ONet, we conduct an experiment. We use pre-trained models of YOLACT and ONet. For CvxNet, instead, we decided to train it on the three categories that are present in both the ShapeNet and the Pix3D datasets: chairs, sofas, and tables. For training CvxNet, we use the data group employed by Choy *et al.* [16]. For the rest of the training details, e.g., the number of hyperplanes, we closely followed the details in the CvxNet paper [17]. We test both the approaches using the raw images and using a version of the images for

which the background was removed using the segmentation from YOLACT. We found that both approaches have achieved an improvement in accuracy when they are executed for background-less (segmented) images. The improvement on a scale of F_1 score is more than 22% for the chair and table, and 12% for the sofa category. A bar chart explaining category-wise improvement in accuracy (due to the background segmentation) is presented in Fig. 3.4. The experiment highlights the significance of the proposed setup (Fig. 3.3) for the 3D reconstruction approaches that are valid only for synthetic datasets.

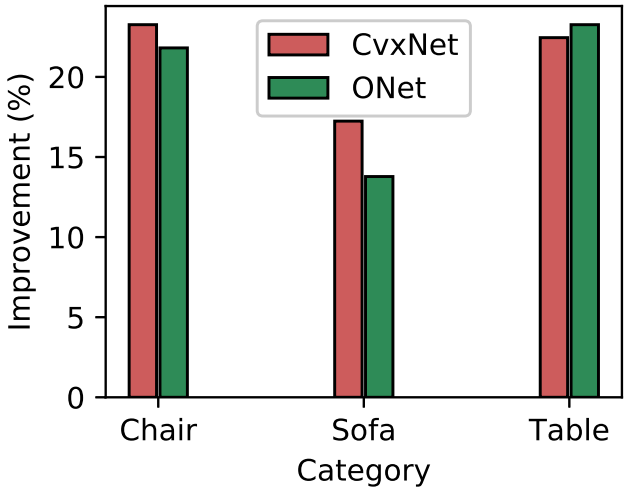


Figure 3.4 Category-wise improvement (on scale of F_1 score) in CvxNet and ONet. Executing the approaches on the masked images produced a much better performance in all cases.

3.3 Results and analysis

We conduct experiments in three different settings. First, we test the SOTA and our approach for white and color background images of the Pix3D dataset. The goal of this experiment is to highlight the significance of the background in 3D shape reconstruction. For this, we divide the dataset into sets; white background images and real (color) background images. Second, we repeat the evaluation for the complete Pix3D dataset without separating the white and color background. Third, we show the qualitative results of the proposed approach for other categories.

3.3.1 Experiment 1: Results for the Pix3D white and real background images

The section presents results to test the generalization, i.e., validation of the proposed approach in different settings. We evaluate the performance on white and color background images. For that, we divide the Pix3D dataset into two groups; white background and color background images. Moreover, in these experiments, both the CvxNet and ONet are trained for the synthetic dataset.

In the first attempt, we conduct an experiment by considering first the group of the test set – real images with a white background. Here we compare reconstructions of CvxNet, ONet, and Mesh R-CNN. Since the images contain only white background, they are treated similar to synthetic ones. Therefore CvxNet and ONet produce good results. The results are illustrated in Fig. 3.5. Although the Mesh R-CNN directly estimates 3D shape from real images, its reconstructions are not comparatively very accurate, smooth and complete. Thus the experiment validates that CvxNet and ONet can produce more accurate results than Mesh R-CNN in simplified (white background) settings.

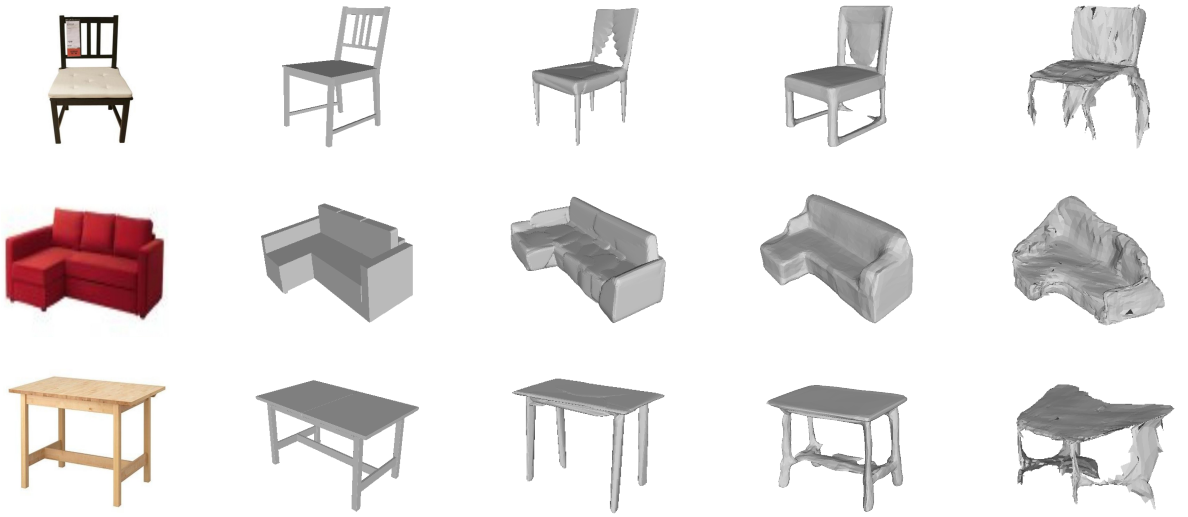


Figure 3.5 Qualitative comparison of CvxNet, ONet and Mesh-RCNN (MR-CNN) for white background Pix3D images.

In the second attempt, we consider the next group of test set containing images with color backgrounds. Initially, the object mask is computed using YOLACT, which helps in separat-

Table 3.1 Quantitative comparison between the baselines and our approach. Columns marked with BG and white correspond to experiments when the color or white background images were used as input. The columns marked as Mask represents results when masked versions of the color images are used. Executing CvxNet and ONet on the masked images produced a much better performance in all cases. The proposed method (ours) achieves better reconstruction accuracy for the sofa and table categories, while CvxNet retains an advantage for masked versions of the table category. The best values for Chamfer L_1 distance and F_1 Score are highlighted in bold.

Category	CvxNet		CvxNet-M	ONet		ONet-M	Mesh R-CNN		Ours	
	BG	White		BG	White		BG	White	BG	White
					F_1 Score (%), τ=0.001 ↑					
Chair	25.04	49.01	**46.30**	23.38	52.31	45.19	37.56	38.06	41.83	48.37
Sofa	37.21	57.30	54.45	38.37	58.40	52.15	51.06	56.75	**61.28**	62.04
Table	22.37	47.44	44.82	20.61	46.19	43.87	45.18	48.42	**48.27**	49.72
Average	28.21	51.25	48.52	27.45	52.30	47.07	44.60	47.74	**50.46**	53.38
					Chamfer L_1 Distance ↓					
Chair	3.52	2.32	1.93	3.13	1.56	**1.54**	2.02	1.85	1.96	1.79
Sofa	2.95	1.51	1.72	2.75	1.73	1.84	1.85	1.72	**1.65**	1.61
Table	6.04	4.21	3.9	5.61	3.81	3.01	2.93	2.63	**2.34**	2.05
Average	4.17	2.68	2.52	3.83	2.37	2.13	2.27	2.07	**1.98**	1.82

ing an object from a background. The separated object is pasted in the center of an image with a white background. The images are used for the evaluation of CvxNet-M and ONet-M. In contrast, the rest of the approaches use images directly without any processing.

An overall quantitative analysis highlighting the performance of the approaches is illustrated in Tab. 3.1. The results are depicted for color (BG) and white background images separately. Based on both metrics (F_1 score and Chamfer L_1 distance), it can be validated that all the approaches perform well on white images and worst on color (BG) images. It is due to the fact that images with a white background are similar to synthetic images for which the models were trained. The table also presents a comparison of the approaches that process input images by applying instance segmentation with the proposed approach. The corresponding columns are highlighted in grey color for straightforward discussion. Both metrics show that our approach performs overall well. In comparison with Mesh R-CNN, its performance is high for every category. However, only for chair category CvxNet-M and ONet-M outperform ours on F_1 score and L_1 distance, respectively. It is due to the fact that most of the images in the chair category contain multiple objects of interest at various positions. In CvxNet-M and ONet-M, the segmenter selects only one object with the highest score; hence the reconstructor

produces a good shape. Whereas, in our approach, the encoder considers all the prominent objects in an image for gathering the features.

3.3.2 Experiment 2: Results for complete Pix3D dataset

In this section, we discuss the results of the baseline approaches and compare them with the results of our approach. CvxNet, ONet, Mesh R-CNN, and the proposed approach are tested using natural images without any pre-processing. However, CvxNet-M and ONet-M compute a masked version of the original image and then reconstruct the 3D shape. All the approaches are tested on the Pix3D dataset. Qualitative results are illustrated in Fig. 3.6. Where input real RGB images, detection of the object from the image, generated masked versions using the detection and the expected 3D shapes for every category are shown in the first three rows, respectively. The next rows illustrate reconstructions by the baselines and the proposed approach (last row). The results of the Cvxnet-M and ONet-M are more accurate than those of CvxNet and ONet. That is due to the fact that they use the segmented foreground part of an image. Reconstructions of the Mesh R-CNN are not complete. In many scenarios, the self-occluded regions are not accurately reconstructed. In comparison, the presented approach outperforms by estimating sharp and smoother surface without requiring any pre-processing on the images.

A quantitative analysis highlighting the performance of the approaches on the Pix3D dataset is illustrated in Tab. 3.2. The CvxNet-M and ONet-M perform well for both metrics in

Table 3.2 Quantitative comparison between the baselines and the approach on the Pix3D dataset. Our approach achieves better reconstruction accuracy on a scale of F_1 score in all the cases, while ONet-M retains an advantage for the masked version of the chair category for Chamfer L_1 distance. The best values are highlighted in bold.

Category	CvxNet	ONet	Mesh R-CNN	CvxNet-M	ONet-M	Ours
	F_1 Score (%), τ=0.001 ↑ / Chamfer L_1 Distance ↓					
Chair	35.43/2.73	34.21/2.45	37.63/1.99	46.88/1.91	46.24/**1.54**	**47.16**/1.82
Sofa	41.99/1.94	42.45/1.91	53.61/1.76	58.35/1.68	53.73/1.75	**61.58/1.63**
Table	33.15/5.07	28.79/5.01	48.12/2.41	44.98/3.72	45.19/2.79	**48.91/2.14**
Average	36.86/3.25	35.15/3.12	46.45/2.05	50.07/2.44	48.39/2.03	**52.55/1.86**

comparison with CvxNet and ONet. This validates that feeding masked images to a 3D reconstruction approach that is trained on synthetic images is beneficial. Second, CvxNet-M and ONet-M show better results than Mesh R-CNN, with the exception of the table class.

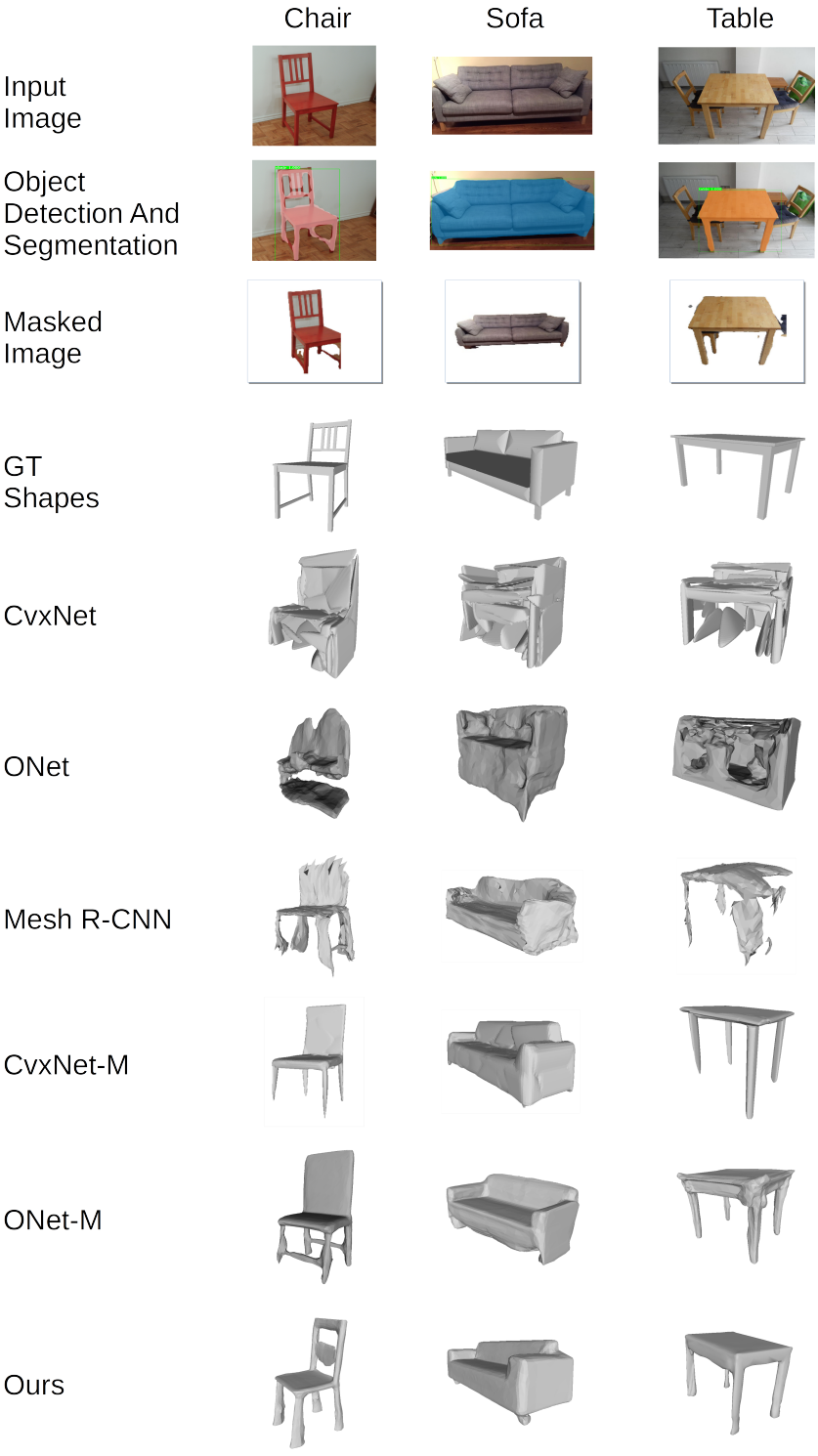


Figure 3.6 Qualitative comparison of the proposed approach (ours) with the baselines. The masked images are obtained by removing the background, centering the object and padding. CvxNet-M and ONet-M use masked images, whereas the rest of the approaches i.e., CvxNet, ONet, Mesh R-CNN (MR-CNN), and ours use natural images.

The problem with the table class originates at the segmentation stage: when the segmentation is not accurate, the 3D reconstruction suffers. In comparison, our approach performs overall well on both metrics. However, for the chair category, ONet-M outperforms ours on Chamfer distance. That is because most of the images in the chair category contain multiple objects of interest and at various positions. In ONet-M, the segmenter selects only one object with the highest score and hence the reconstructor produces a good shape. Whereas, in our approach, the encoder considers all the prominent objects in an image for gathering the features.

For testing on realistic images, we train CvxNet, ONet, and our approach on the realistic dataset. The results are presented in Table 3.3. Although the CvxNet and ONet are trained on the realistic dataset, they still could not perform well on the test set. That is due to the fact that they consider background for feature learning, which is variant. On the other hand, our approach performs well as it is trained to extract object features in the presence of the background.

Table 3.3 Quantitative results for the realistic dataset. The performance of our approach is comparatively better than CvxNet and ONet on both metrics.

Category	Chair	Sofa	Table	Airplane	Bench	Phone	Display	Vessel	Car	Avg.
					F_1 Score (%), τ=0.001 ↑					
CvxNet	49.13	62.39	49.87	71.21	54.23	62.93	47.10	53.20	50.83	55.65
ONet	53.21	68.02	62.32	58.76	60.72	66.78	40.82	47.20	64.33	58.02
Ours	**58.32**	**71.23**	**65.93**	**73.30**	**62.43**	**68.11**	**52.21**	**54.74**	**66.32**	**63.22**
					Chamfer L_1 Distance ↓					
CvxNet	1.83	0.87	1.82	1.01	1.32	1.19	1.95	0.92	0.87	1.31
ONet	1.52	0.91	0.83	1.22	1.58	1.27	2.06	1.06	0.92	1.26
Ours	**1.48**	**0.78**	**0.71**	**0.93**	**1.21**	**1.03**	**1.89**	**0.87**	**0.83**	**1.08**

3.3.3 Experiment 3: Quantitative results for other categories

In order to highlight the robustness of the proposed approach and the presented setup for CvxNet and ONet, we present reconstruction results for some other categories in Fig. 3.7. For every sample, we show two views of the reconstructed shapes. It can be observed that CvxNet and ONet could not produce reasonable 3D shapes for real images. However, for masked images, their reconstructions are quite good. The last column depicts shapes reconstructed by our approach. The approach shows promising results by accurately estimating the boundary of the objects without requiring any pre-processing on the test images.

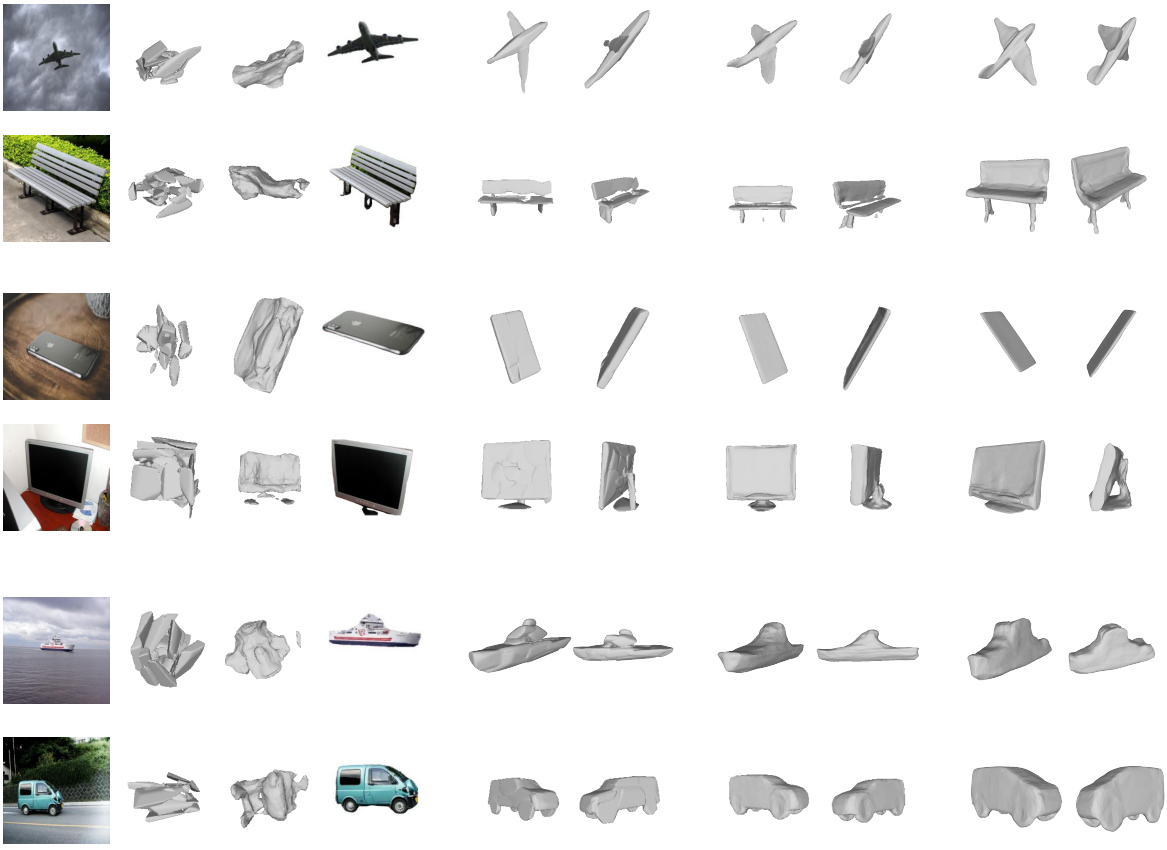


Figure 3.7 Qualitative comparison among CvxNet-M and ONet-M and our approach. CvxNet-M and ONet-M are evaluated for the masked versions of the input images. In comparison, the proposed approach (ours) uses input images directly without any pre-processing.

3.4 Ablation study

Comparison of YOLACT and Mask R-CNN: We consider Mesh R-CNN as one of the baselines that computes instance segmentation in the early stage using Mask R-CNN. On the other hand, we utilize YOLACT for the same purpose in CvxNet-M and ONet-M, as discussed in Sec. 3.2.2. These three baselines are dependent on a segmentation method, therefore, for unbiased experiments, we first compare the performance of YOLACT and Mask R-CNN. Their performance on the selected categories of the Pix3D dataset is reported in Tab. 3.4. We find that YOLACT can segment a smaller portion of images than Mask R-CNN. The difference is large for the table category because YOLACT is trained to detect dining tables, so it does not perform well on other kinds of tables.

Table 3.4 Missed detection by YOLACT and Mask R-CNN on the Pix3D dataset. YOLACT's performance is worse, especially for the table category.

Pix3D Test Set		Missed Detection (%)	
Category	Images	YOLACT	Mask R-CNN
Chair	3839	4.87	0.16
Sofa	1947	5.55	0.05
Table	1870	39.36	0.21
Overall	7656	13.47	0.14

For the comparison between our and the SOTA approaches (as presented in this chapter), only those images are selected for which both YOLACT and Mask R-CNN produce a segmentation. The number of images for each category was thus reduced. Furthermore, by taking into account the nature of the considered problem, we divided the test set into two groups based on their background. All the images with the white background are added into the first group, whereas the second group contains the rest of the images with color backgrounds. Tab. 3.5 depicts the quantity of images in every category of the reduced test set.

Table 3.5 Test set with color and white background (BG) group

Category	Color BG	White BG	Total
Chair	3163	468	3631
Sofa	1375	461	1836
Table	1005	105	1110
Overall	5543	1034	6577

3.5 Chapter summary

The objective of this chapter is to reconstruct 3D shapes from a single real image. In this regard, an end-to-end approach is proposed that strives to extract object features from a real image by reducing the influence of the image background. During training, a synthetic image is fed to the encoder with its realistic version. The encoder extracts common features from both images that represent features of the object. The extracted features are used by the model to estimate the object's 3D shape. During inference, we test on real images. The proposed approach outperforms SOTA approaches which are validated by conducting a series of experiments. Furthermore, a baseline system is designed that enables CvxNet and ONet to extract accurate 3D shapes from real images. That system entails segmenting the object of interest from the input images and removing the image background before passing them to the reconstruction algorithms.

The proposed solutions estimate the 3D shape in the canonical pose irrespective of the pose of an object in the input image. However, for many real-world applications, such as the interaction of an articulated robot with an object, require knowledge of the object's pose. In the next chapter, we present a method to estimate the 3D keypoints of an object from an RGB image, which can be used to compute the pose information.

Chapter 4

3D Keypoints Estimation from A Single RGB Image

This chapter introduces a method to estimate keypoints in 3D space from single-view RGB images. The 3D keypoints preserve an object's structural information; shape, position and orientation, which are required for solving several scene-understanding tasks, including object detection and matching, geometrical reasoning, human-robot interaction, manipulation, navigation in cluttered environments, path planning, etc.

Recent research has shown that these tasks can be addressed using keypoints [4, 51, 142] as they represent Points of Interest (PoI) which are invariant to transformations including rotation, translation, scaling, etc. [96, 1, 5, 55, 91, 93, 98]. Moreover, the ordered list of semantic keypoints could be helpful in finding correct correspondences between the points in two images (using 2D keypoints) [140], point clouds or meshes (using 3D keypoints) [130, 22, 90, 102, 130, 135]. Also, the correspondences between the 2D keypoints from multi-view images can help estimate the depth [57].

Most of the recent studies use 3D keypoints for various human-related applications including joint detection, motion capturing, pose estimation, etc., which deal with a single category (human) and a fixed number of keypoints [58, 101, 73, 134, 106, 105, 50, 72]. On the other hand, keypoints are also used in applications related to rigid objects (i.e. cars, chairs, etc.), where an object's structure and the number of keypoints may vary depending on the category. To simplify the problem, the existing approaches train their network separately for every category for a fixed number of keypoints.

In the literature, the existing approaches for 3D keypoint estimation select the keypoints from the given 3D data using classification techniques. This rationale inherently constrains the selection of points only from those lying on the given object and, therefore, performing overall well [135, 15, 89, 128, 113, 6]. Differently, the problem becomes more challenging when the 3D keypoints position has to be estimated using single-view RGB images [66, 98, 145]. This is because the problem is severely ill-posed, and the estimated points may not necessarily lie on the object in 3D, even if visualised correctly on the 2D image plane. Therefore, in general, the image based approaches either estimate 2D keypoints $[u, v]$ for two views of the same object and then use triangulation to estimate the depths $[d]$ [98], or estimate 2D keypoints using RGB, RGBD and/or silhouettes and project them to the object's 3D shapes in order to find the 2D-3D correspondences [28]. Since such approaches use 2D keypoints to estimate depths which do not guarantee accuracy, their performance is not as good as the point cloud based methods [37].

In comparison, we present an approach that uses a single-view RGB image and estimates an ordered list of keypoints in 3D space. Moreover, keypoints estimated by our approach are equivariant (not in canonical pose), i.e., the pose of the keypoints is the same as the pose of an object in the input image. An overview of our approach is shown in 4.1, highlighting the key difference with the existing approaches.

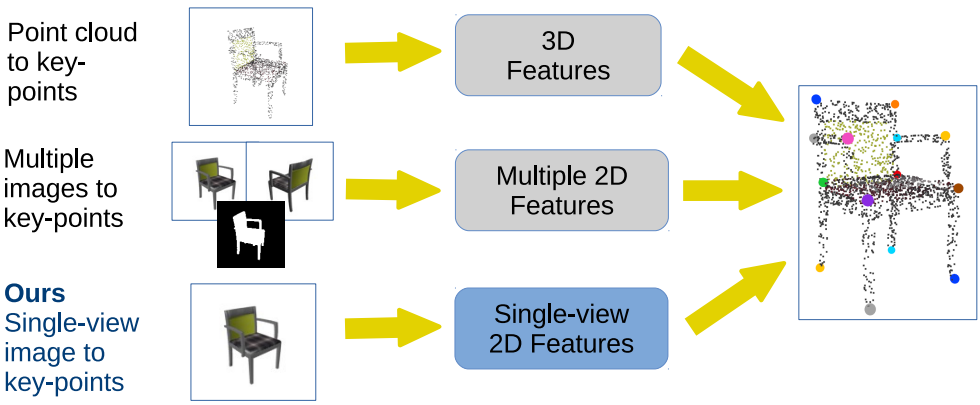


Figure 4.1 Comparison with the other paradigms. Some existing methods use point clouds (top) or multiple images representing different views of an object (middle) as inputs and compute 2D/3D features for keypoints estimation. In comparison, the proposed approach considers a single-view RGB image, extracts object 2D features, and use them for estimating 3D keypoints (bottom).

Our main contributions are as follows;

- The proposed approach estimates keypoints from a single-view RGB image.

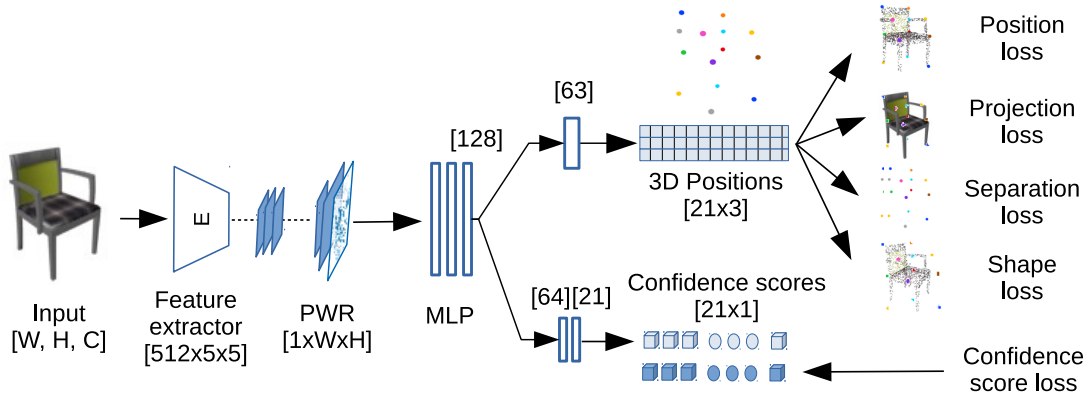


Figure 4.2 The proposed architecture. An RGB image is fed to a feature extractor to produce object features that are up-sampled in order to achieve a Pixel-wise Representation (PWR). Finally, a Multilayer Perceptron (MLP) is added that uses PWR for estimating 21 keypoints in 3D space along with confidence scores.

- Unlike the existing approaches, the proposed approach estimates a confidence score for every keypoint, which allows it to select valid keypoints from the set of estimated keypoints.
- The estimated keypoints provide order-wise semantic information that is independent of the object's view.
- Our method is a flexible approach that can predict a geometry-based number of keypoints, to accommodate inter-and-intra-class shape variations.
- The approach can be trained for various categories simultaneously and is capable of estimating keypoints of self-occluded parts of the objects.
- The estimated keypoints can be used for downstream tasks such as shape alignment and relative pose estimation between two objects.

4.1 Methodology

Given an RGB image, our work aims at estimating an ordered list of 3D keypoints that are semantically and geometrically consistent across different instances of an object category. For this, an end-to-end approach is proposed that extracts an object's features from an image, computes a Pixel-wise Representation (PWR) and uses the representation for the estimation of 3D keypoints along with confidence scores. The architecture of the approach is illustrated in Fig. 4.2.

The presented approach is based on three modules. The first module (feature extractor) takes an RGB image as input and produces feature vectors. These extracted features are converted to PWR in the second module. The PWR has the same width and height as the input image. However, instead of representing the RGB value, every pixel represents a feature for the corresponding pixel of the input image. The third module contains a Multi-Layer Perceptron (MLP) based on four linear layers. The PWR features are flattened to a 1D tensor before feeding to the MLP. The MLP uses them for estimating 21 keypoints. For every keypoint, a position in 3D space $[x, y, z]$ and a confidence score (from 0 to 1) is computed. The confidence score reflects how confident the network is that the keypoint exists for the object. If such a value is greater than 0.5, it means that the predicted keypoint exists for the object, and it is considered as a valid keypoint. Otherwise, it is discarded. In this way, the network selects an object's valid keypoints from the predicted 21 keypoints. So, the total number of valid keypoints could be different for different shapes of objects.

The network is trained separately for every category (as followed in literature) as well as jointly for all the categories. We found that the results with both methods are approximately the same. So, we report the results of a network trained jointly for all the categories. The network minimizes five losses: 3D position loss, 2D projection loss, separation loss, shape consistency loss and the confidence score loss.

Estimation of the keypoints 3D postions

Considering N to be an upper bound for the number of keypoints across all the shape classes, the valid keypoints for every object could be less than or equal to N. For example, objects in the *cars* category would, in general, have a different number of valid keypoints than those in the *chairs* category. Moreover, there can also be intra-class variations. In other words, the valid keypoints are those that exist for an object and, therefore, vary with geometry.

In the ground truth, keypoints are arranged in an ordered list. Consider a set $C_Q = \{cq_1, \ldots, cq_N\}$ representing keypoint validity in the ground truth such that $cq_k = 1$ if keypoint exists at index k else $cq_k = 0$. If the object contains M valid keypoints, then we have $\sum_{k=1}^{N} cq_k = M$. We then create two sets $\mathscr{P} = \{p_k | k = 1, ..., N \; if \; cp_k == 1\}$ and $\mathscr{Q} = \{q_k | k = 1, ..., N \; if \; cp_k == 1\}$ containing the predicted and ground truth 3D positions, respectively, for the valid keypoints. Here, the cardinality of the sets is $|\mathscr{P}| = |\mathscr{Q}| = M$.

Using the above notations, the **3D position loss** that measures the accuracy of the predicted 3D positions is computed using the Mean Square Error (MSE) as:

$$\mathscr{L}_{pos} = \frac{1}{M}\sum_{i=1}^{M}\left\|p_i - q_i\right\|_2^2. \tag{4.1}$$

where $p_i \in \mathscr{P}$ and $q_i \in \mathscr{Q}$ are the corresponding predicted and ground truth 3D positions, respectively. In order to predict a more accurate position of the keypoints, we also compute the loss in 2D space. To do so, both, the valid estimated (p_i) and ground truth (q_i) keypoints are transformed from 3D ($[\bar{x}_i, \bar{y}_i, \bar{z}_i]^\top$ and $[x_i, y_i, z_i]^\top$) to 2D ($[\bar{u}_i, \bar{v}_i]^\top$ and $[u_i, v_i]^\top$) pixel coordinates using the known transformation P (camera intrinsic and extrinsic) [98]. For the **2D projection loss** ($\mathscr{L}_{proj}$) we take the Mean Absolute Error (MAE) between estimated and ground truth 2D pixels as:

$$\begin{gathered} p_i = [x_{p_i}, y_{p_i}, z_{p_i}]^\top, \;\; q_i = [\bar{x}_{q_i}, \bar{y}_{q_i}, \bar{z}_{q_i}]^\top, \\ [u_i, v_i]^\top = P(p_i), \;\; [\bar{u}_i, \bar{v}_i]^\top = P(q_i), \\ \mathscr{L}_{proj} = \frac{1}{M}\sum_{i=1}^{M}\left\|[u_i, v_i]^\top - [\bar{u}_i, \bar{v}_i]^\top\right\|, \end{gathered} \tag{4.2}$$

Without any additional constraints, the network can predict more than one keypoint at the same 3D location, which is not realistic. To address this, we penalize the condition when the Euclidean distance between the predicted keypoints is less than a pre-defined hyperparameter δ^2 using the **separation loss** ($\mathscr{L}_{sep}$) defined as:

$$\mathscr{L}_{sep} = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j\neq i}^{M}\max(0, \delta^2 - \left\|p_i - p_j\right\|_2^2), \tag{4.3}$$

where p_i and p_j represent the i^{th} and j^{th} predicted keypoints, respectively. We use $\delta^2 = 0.05$ in our implementation to have a sufficiently large separation between the predicted keypoints.

Unlike the existing point cloud based approaches, in case of image based approach, the keypoints may also be predicted in the object's surrounding (or on the image background in case of 2D predicted keypoints). We overcome this limitation by introducing a **shape consistency loss** ($\mathscr{L}_{shape}$) that forces the network to estimate keypoints closer to the object's surface. The loss minimizes the distances of the predicted keypoints from their nearest

neighbor points in the ground truth point clouds. The loss can be described as:

$$\begin{aligned} d_i &= \left\| p_i - kNN(p_i, \mathscr{PC}) \right\|_2, \\ \mathscr{L}_{shape} &= \frac{min(1,C)}{max(1,C)} \sum_{i=1}^{M} d_i, \quad \text{if } d_i > \gamma, \end{aligned} \tag{4.4}$$

where, $kNN()$ is a function that finds the nearest neighbor of a valid predicted keypoint p_i from a point cloud $\mathscr{PC}$ of an object which is available during the training time. The value d_i is the distance of a keypoint p_i from its nearest neighbor point. C is a count of considered distances that are greater than the threshold (γ). It is used as a way to average the distances, and when C is zero, the $\mathscr{L}_{shape}$ is set to zero. We select γ as 0.05 that represents a tolerance distance. The final loss ($\mathscr{L}_{shape}$) is an average of the considered distances.

Estimation of confidence scores

Confidence scores play a vital role in the proposed approach because the total number of valid keypoints may not be the same for all the objects. During training, the ground truth information about the valid and invalid keypoints is available in the set C_Q discussed earlier. However, since the ground truths are not available during inference, it becomes very challenging for a network to identify the valid keypoints from the predicted ones.

The existing approaches solve this problem by either estimating a fixed number of keypoints for all the objects in the dataset or by training their model separately for every category (by keeping fixed keypoints for all the objects of a category). Their solutions are not ideal because of two reasons: 1) fixing the number of keypoints may not accurately represent objects of different shapes of the same category i.e., two chairs of different structures could require a different number of keypoints for representing the shape more precisely, 2) training a network separately for every category is not a generalized solution, as such networks may work well only for those objects that have a geometric structure similar to the ones used in training. In comparison, we solve the problem of identifying valid keypoints, by estimating a confidence score for every predicted keypoint.

Let $C_P = \{cp_1, \ldots, cp_N\}$ represent the predicted confidence scores. We then compute the **confidence score loss** ($\mathscr{L}_{conf}$) as:

$$\mathscr{L}_{conf} = \frac{1}{N} \sum_{k=1}^{N} \left\| cp_k - cq_k \right\|, \tag{4.5}$$

where $cp_k \in C_P$ could range from 0 to 1, and $cq_k \in C_Q$ could be either 1 or 0, representing the keypoint validity in the ground truth.

So the overall network loss ($\mathscr{L}_{overall}$) can be defined as a weighted sum of all the above losses as:

$$\mathscr{L}_{overall} = \alpha_{pos}.\mathscr{L}_{pos} + \alpha_{proj}.\mathscr{L}_{proj} + \alpha_{sep}.\mathscr{L}_{sep} + \alpha_{shape}.\mathscr{L}_{shape} + \alpha_{conf}.\mathscr{L}_{conf}. \quad (4.6)$$

In order to balance the effect of every loss $[\alpha_{pos}, \alpha_{proj}, \alpha_{sep}, \alpha_{shape}, \alpha_{conf}]$ are selected as [1, 0.33, 1, 1, 1], respectively.

Inference
During inference, the network predicts 21 3D keypoints along with their confidence scores from a single image. All the keypoints having a confidence score greater than 0.5 are selected as valid keypoints. The rest of the keypoints are discarded. For better visualization, the predicted valid keypoints are illustrated on the original point cloud of the object (i.e. 4.4).

4.2 Experimental setup

The section presents the experimental details, an arrangement of the dataset, and explains metrics selected for performance evaluation.

4.2.1 Implementation details

The feature extractor module is based on ResNet-18 that is pre-trained on ImageNet dataset [18]. We discard its last two layers to extract features of dimensions 512x5x5. The network is implemented in PyTorch and trained with Adam optimizer. The learning rate is 10^{-3}, and the batch size is 512.

4.2.2 Dataset

As extensively evaluated in previous approaches, we use the KeypointNet dataset [130] to analyse the performance of our approach. We load pairs of the images in random views and corresponding ground truth 3D keypoints. The images are fed to the proposed network to estimate 3D keypoints in the same pose as the pose of input image. The 3D ground truth keypoints are used to evaluate the accuracy of the estimated keypoints.

4.2.3 Performance measurement

We compare our results with those of KP-Net [98]. Unlike the existing point cloud based methods [130, 24, 41, 11, 117], their approach (in inference) uses single image and estimates 3D keypoints (pixel $[u,v]$ and depth $[d]$). It estimates 3D keypoints for two views of an object. The keypoints are then used for finding a pose (rotation matrix) between the object views. The estimated pose is compared with the ground truth pose by computing an angular distance error.

We follow the same procedure and estimate keypoints for two views (A and B) of an object using our approach. However, for evaluation, we use these keypoints in two different methods. The first method is exactly the same as KP-Net, where we compute the relative rotation matrix ($\bar{R}$) between object views using Procrustes analysis and then calculate the angular distance error (E_T) between computed and ground truth relative rotation matrix (R) as:

$$E_T = 2\, arcsin\left(\frac{1}{2\sqrt{2}}\,||\bar{R}-R||_F\right), \tag{4.7}$$

where $||.||_F$ is a Frobenius norm.

As a second evaluation, we transform the estimated keypoints of view A ($\mathscr{A} = \{a_i|i = 1,...,M\}$) using the predicted ($\bar{R}$) and the ground truth ($R$) rotation matrix and call them $\mathscr{A}_p = \{ap_i|i = 1,...,M\}$ and $\mathscr{A}_q = \{aq_i|i = 1,...,M\}$, respectively. Generally, both the keypoints $\mathscr{A}_p$ and $\mathscr{A}_q$ should lie on the same positions as the keypoints of view B (see Fig. 4.4). Every keypoint ap_i/aq_i of $\mathscr{A}_p/\mathscr{A}_q$ is considered as a vector from the origin ($\mathbf{ap}_i, \mathbf{aq}_i$). An angular distance error (E_P) between $\mathscr{A}_p$ and $\mathscr{A}_q$ is computed using vector dot product as:

$$E_P = \frac{1}{M}\sum_{i=1}^{M} arccos\left(\frac{\mathbf{ap}_i \cdot \mathbf{aq}_i}{|\mathbf{ap}_i|\,|\mathbf{aq}_i|}\right), \tag{4.8}$$

where M is the total number of estimated valid keypoints. For a fair comparison with the KP-Net, we consider the first evaluation. Nevertheless, for validation on other categories, results from both evaluations are presented.

4.3 Results and Analysis

This section evaluates the performance of the proposed approach. First, we present the results of our approach for those categories of the KeypointNet dataset that are not tested by the KP-Net [98]. Second, we test our approach for the three categories considering [98]

(airplane, car and chair) and compare the results with those of the KP-Net. Third, we show the significance of the confidence scores by computing the results for the keypoints selected based on the estimated confidence scores with those of ground truth confidence scores.

4.3.1 Performance of the proposed approach

The proposed approach is evaluated using images with white background of 13 different categories of the KeypointNet dataset – 10 more than the KP-Net [98]. For that, two views of the same object are passed to the network for estimating 3D keypoints for every view. The Procrustes analysis is used that utilizes the estimated keypoints to compute a relative pose (rotation matrix) between the keypoints estimated in the two views. The estimated relative pose is compared with the original pose between in the input images (ground truth pose) to compute the angular distance errors using the defined evaluation metrics (Eq. 4.7 and Eq. 4.8). The angular distance errors are depicted in Tab. 4.1.

Table 4.1 Error in pose estimation between two views of an object. Angular distance error is computed in degrees between; 1) estimated and ground truth rotation matrices (Eq. 4.7) and 2) 3D positions (Eq. 4.8) of the predicted keypoints in two views. This experiment is conducted for white background images.

Category	E_T		E_P	
	Mean	Median	Mean	Median
Airplane	6.581	3.145	5.963	2.565
Car	6.761	2.980	5.316	2.456
Chair	13.562	5.017	11.247	4.566
Table	23.919	3.635	18.079	2.975
Vessel	14.652	4.392	11.655	3.478
Bed	28.598	12.422	25.332	9.049
Cap	16.904	8.193	13.634	6.261
Helmet	26.947	16.058	23.504	15.243
Knife	25.330	13.006	20.599	12.490
Motorcycle	9.467	3.226	6.490	2.507
Guitar	19.559	5.289	7.247	2.926
Mug	18.470	9.135	10.320	5.942
Bottle	17.118	14.854	14.674	12.013
Average	16.962	7.690	12.822	6.190

Table 4.2 MSE is computed between the 3D positions of the predicted and ground truth keypoints. Consider the maximum error as $\sqrt{3}$, the error for the estimated keypints is very small for all the categories. This validates that the keypoints are estimated very close to the ground truth keypoints.

Category	MSE	
	Mean	STD
Airplane	0.006	0.017
Car	0.008	0.040
Chair	0.015	0.049
Table	0.053	0.159
Vessel	0.026	0.075
Bed	0.094	0.163
Cap	0.031	0.063
Helmet	0.062	0.076
Knife	0.008	0.006
Motorcycle	0.011	0.045
Guitar	0.003	0.006
Mug	0.026	0.056
Bottle	0.023	0.027
Average	0.028	0.060

The error is comparatively high for some categories. That is due to the structural variation (single/bunk beds, tables), different keypoints for similar object shapes (helmet, knife, etc.), and differences in center of rotation and the center of mass of the object (i.e., mug).

Furthermore, to evaluate the keypoints positions to show how accurately the keypoints are localized with respect to the positions of the ground truth keypoints. To do so, we compute the Mean square error between the estimated and the ground truth keypoints' 3D locations. In Tab. 4.2 we show the mean and Standard Deviation (STD) of the error (MSE) between predicted and ground truth 3D keypoints. Since the keypoints are normalized in a unit volume, the maximum error (distance between the keypoints) could be $\sqrt{3}$. In comparison, it can be seen that the error is very small. This validates that the keypoints are estimated very close to the ground truth keypoints.

Qualitative results are depicted in Fig. 4.3. The first and the fourth row show the test images, the second and the fifth row present the estimated keypoints on top of the original point clouds of the objects, and corresponding ground truth keypoints are shown in the third and the sixth row. It can be observed that the keypoints are estimated approximately on valid 3D positions and are in semantic order with respect to the ground truth keypoints. Moreover, the proposed approach is able to predict 3D keypoints for the occluded parts of the objects. For example, one leg of the table and the bed is not visible in the images because of self-occlusion. However, keypoints are accurately estimated for them.

4.3.2 Comparison with KP-Net

To compare our results with the KP-Net, we consider the same three categories (cars, airplanes, and chairs) as reported in [98]. Moreover considering the same settings, we use transparent images in this experiment. For the evaluation, we compute the angular distance error between the estimated and ground truth pose of two views of an object using only the Eq. 4.7 (as used by KP-Net). The error is depicted in Tab. 4.3. In the KP-Net, authors have presented results for four different versions of their approach; 1) supervised KP-Net that learns from ground truth 2D pixels and corresponding depths, 2) supervised KP-Net with a pretrained Orientation Network (O-Net) that provides an object's orientation information, 3) KP-Net (unsupervised) with O-Net, and 4) KP-Net without O-Net. It is reported that the KP-Net without O-Net performs overall well. The first four rows of Tab. 4.3 present results of the four versions of the KP-Net. The fifth row shows the results of the proposed approach. The lower values show better results. It can be observed that our results are more accurate than those of the KP-Net.

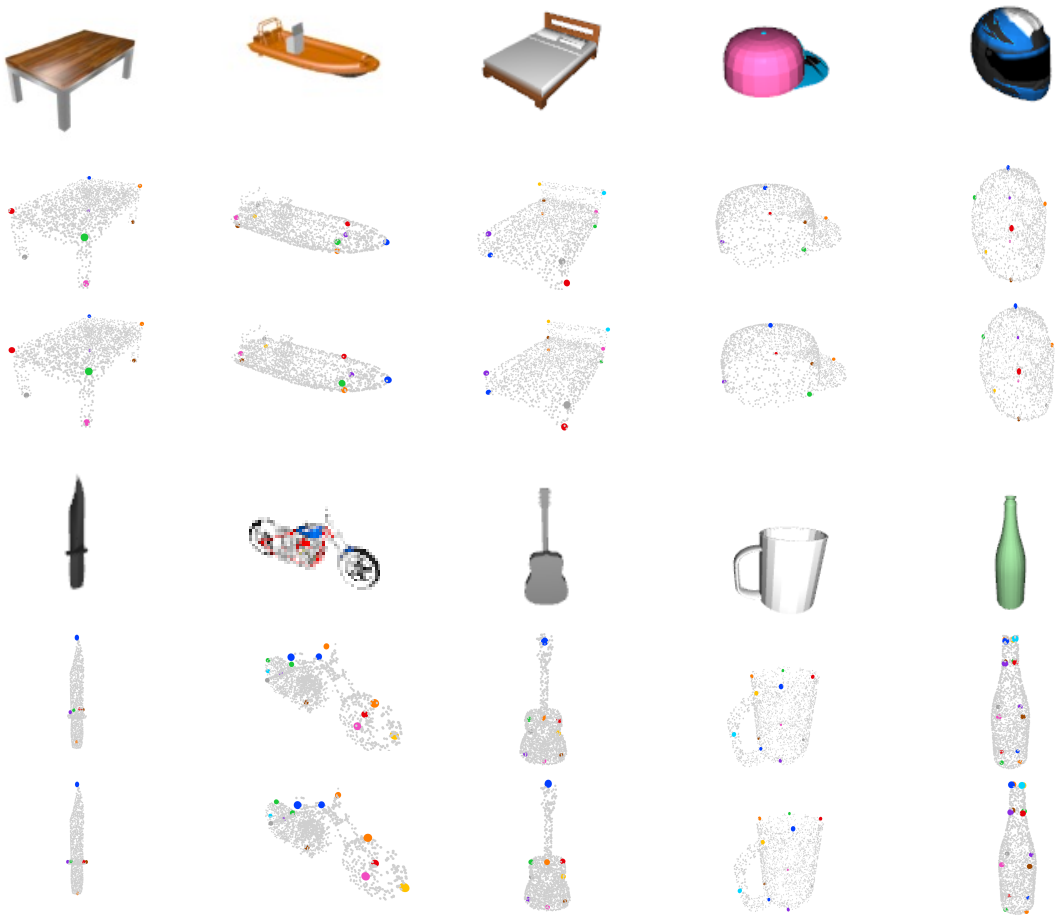


Figure 4.3 Qualitative results of the proposed approach for other ten categories. Row (1, 4) show the input images, row (2, 5) and row (3, 6) present the corresponding estimated and ground truth keypoints, respectively. It can be visualized that the proposed approach estimates a semantically ordered list of keypoints even for the occluded parts of the objects.

Table 4.3 Error in pose estimation between two views of the same object. Mean and median angular distance errors are calculated (in degrees) between ground truth rotation and the rotation computed by Procrustes estimates between predicted keypoints of the two views. Results of the baselines (first four rows) are the same as reported in [98]. All the results are produced for transparent images.

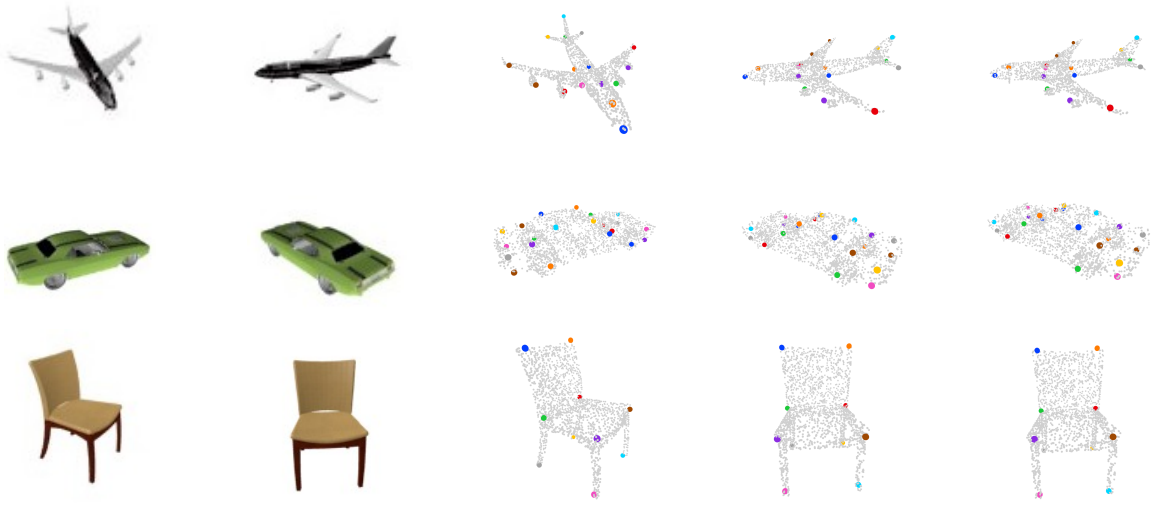
Method	Car		Airplane		Chair	
	Mean	Median	Mean	Median	Mean	Median
Supervised KP-Net	16.268	5.583	18.350	7.168	21.882	8.771
Supervised KP-Net with O-Net	13.961	4.475	17.800	6.802	20.502	8.261
KP-Net with O-Net	13.500	4.418	18.561	6.407	14.238	5.607
KP-Net	11.310	3.372	17.330	5.721	14.572	5.420
Ours	**5.190**	**2.073**	**3.257**	**2.053**	**10.732**	**4.096**

Qualitative results are illustrated in Fig. 4.4. Columns (a) and (b) show two views of the same object. The corresponding estimated keypoints are presented in columns (c) and (d), respectively. Finally, the keypoints (and point clouds) of view A after transformation using estimated (A_{est}) rotation are illustrated in (e). It can be visualized that the pose of the transformed keypoints (e) is the same as the pose of keypoints of view B (d). The experiment highlights that; 1) the estimated keypoints can be used for computing a pose between two views, 2) the keypoints are in semantical order, which is independent of the object view, and 3) the network can predict keypoint of the occluded part of the object (i.e., back legs of the chair).

4.3.3 Significance of confidence score

Furthermore, we present another experiment that highlights the significance of the confidence score. We compare the predicted valid keypoints (to whom the network assigns confidence greater than 0.5) with the keypoints known to be present because of ground truths. The results are approximately the same in both cases, which validates that the confidence score helps the network in classifying the valid keypoints for every object. The results are given in Tab. 4.4 and Tab. 4.4 for RGB and RGBA images, respectively. The tables show the mean angular distance error and the Standard Error (SE), which is calculated as $\sigma/\sqrt{n}$, where σ is the standard deviation of n angular distance errors.

In a nutshell, it can be inferred that if a network could not estimate the confidence scores, it should predict fixed numbers of keypoints as followed by the existing approaches. Otherwise,



(a) View A (b) View B (c) KP A (d) KP B (e) KP A_{est}

Figure 4.4 Computing pose between two views (a) and (b) of an object. The corresponding estimated keypoints are shown on the original point clouds in (c) and (d). The keypoints of view A (c) are transformed to view B using estimated truth rotation matrix as illustrated in (e).

Table 4.4 Results for white background images (RGB). Comparison of the keypoints predicted as valid by our network based on confidence scores (Pred.) with the keypoints selected using ground truths (GT). The pose estimation error in two views of an object is approximately the same in both cases; either the Pred. or GT keypoints are used. Mean and SE of the pose error (calculated in both the methods using (a) rotation matrices (Eq. 4.7) and (b) keypoints 3D positions (Eq. 4.8)).

Category	Metric	E_T		E_P	
		Pred.	GT	Pred.	GT
Airplane	Mean	3.257	3.267	2.805	2.797
	SE	0.075	0.076	0.001	0.001
Car	Mean	5.190	5.187	4.040	4.057
	SE	0.277	0.280	0.004	0.004
Chair	Mean	10.73	10.71	7.53	7.54
	SE	0.330	0.327	0.006	0.004

Table 4.5 Results for transparent images (RGBA). Comparison of the keypoints predicted as valid by our network based on confidence scores (Pred.) with the keypoints selected using ground truths (GT). The pose estimation error in two views of an object is approximately the same in both the cases; either the Pred. or GT keypoints are used. Mean and SE of the pose error (calculated in both the methods using (a) rotation matrices (Eq. 4.7) and (b) keypoints 3D positions (Eq. 4.8))

Category	Metric	E_T		E_P	
		Pred.	GT	Pred.	GT
Airplane	Mean	6.581	6.552	5.963	5.974
	SE	0.195	0.194	0.003	0.003
Car	Mean	6.761	6.764	5.316	5.334
	SE	0.318	0.318	0.004	0.004
Chair	Mean	13.56	13.56	11.25	11.25
	SE	0.340	0.340	0.004	0.004

It may not be possible for the network to separate valid keypoints from the total predicted N (21) keypoints. Moreover, the confidence score allows jointly training a network for several categories with a different number of keypoints. Otherwise, either the network can be trained for a single category, or the total keypoints should be fixed for all the categories.

4.4 Ablation study

4.4.1 Network without the PWR module

We revise the experiments for the transparent images of the three categories by removing the PWR module from the proposed network. The network can still attain better results than KP-Net [98]. However, the accuracy has reduced slightly in comparison with the complete network (with the PWR module). The comparison is shown in Tab. 4.6.

Table 4.6 Results for the architecture with and without the PWR module

Method	Cars		Planes		Chairs	
	Mean	Median	Mean	Median	Mean	Median
Ours with PWR	**5.190**	**2.073**	**3.257**	**2.053**	**10.732**	**4.096**
Ours without PWR	6.293	2.538	4.924	2.860	13.569	5.721

4.4.2 Test for realistic images

We evaluate our approach for images with real backgrounds that are taken from SUN dataset [119]. The angular distance error (in degrees) in pose estimation between two views of an object is depicted in Tab. 4.7.

Table 4.7 Results of our approach for images with a real background. The angular distance errors are calculated in degrees between the predicted and the ground truth rotation matrix using Eq. 4.7.

Method	Car		Airplane		Chair	
	Mean	Median	Mean	Median	Mean	Median
Ours with real background	41.47	12.84	51.01	29.27	70.782	61.52

Qualitative results of the proposed approach for realistic images are shown in Fig. 4.5. Column (a) shows the input images, whereas columns (b) and (c) depict the estimated and the corresponding ground truth keypoints. The experiment shows that the results of real images are much worse than those of synthetic images; RGB with a white background or RGBA with transparent background. That is due to the fact that the network could not separate the object from the background and hence estimates some keypoints in the surrounding. This is our future task to improve the network for estimating more accurate 3D keypoints from images with real background.

4.4.3 Distribution of angular distance error

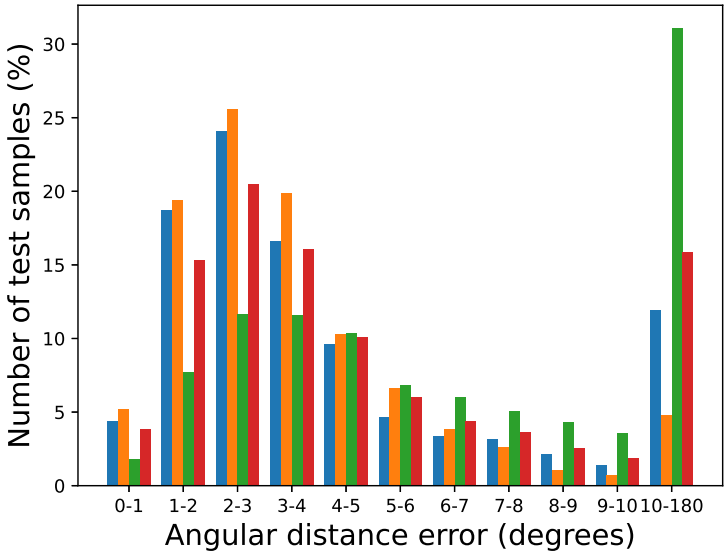
We present the distribution of the angular distance error computed between predicted and ground truth rotations (using Eq. 4.7). We consider RGB (white background) and RGBA (transparent background) images. The corresponding histograms representing the computed distributions are shown in Fig. 4.6. It is observed that in both the cases RGB (Fig. 4.6a) and RGBA (Fig. 4.6b), the error for most of the test samples lies within 0 to 5 degrees. The error is less when RGBA images are used. Moreover, the error is comparatively high for the airplane category.

4.5 Chapter summary

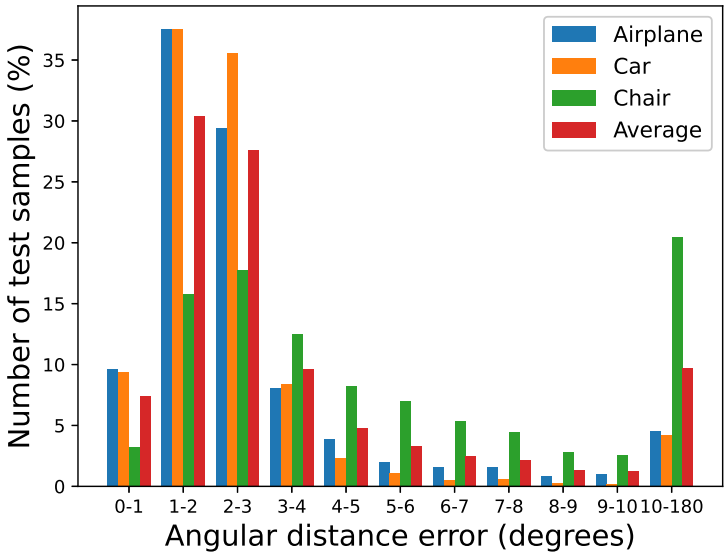
The chapter presents an end-to-end solution for 3D keypoints estimation from a single-view RGB image. The proposed approach extracts object features from an image, computes

(a) Input Image (b) Estimated keypoints (c) GT keypoints

Figure 4.5 Qualitative results of our approach for realistic images. (a) shows test images containing an object with a random background, (b) and (c) illustrate predicted and corresponding ground truth keypoints on the object's point cloud, respectively.



(a) RGB images



(b) RGBA images

Figure 4.6 Distribution of angular distance error calculated between predicted and ground truth rotations computed using Eq. 4.7. (a) and (b) show results for RGB (white background) and RGBA (transparent) images, respectively.

pixel-wise features by upsampling, and uses them for estimating 3D keypoints along with confidence scores that reflect the validity of the keypoints. It enables the network to predict a different number of keypoints based on the object's shape. The keypoints are estimated in an

ordered semantic list, which increases its significance. Moreover, the network can be trained together for all the classes. The approach is evaluated by computing the pose between two views of an object. The presented results show that the proposed approach outperforms the SOTA approaches.

We observed that the presented approach learns to estimate keypoints only from images without considering any 3D information as input. The estimations can be improved by utilizing complete point clouds of the objects. As an extension of the approach we have presented in this chapter, in the next chapter, we propose a teacher-student network that leverages point cloud data during training to estimate the keypoints. During inference, the network estimates the same keypoints from only RGB images.

Chapter 5

CDHN: Cross-Domain Hallucination Network For 3D keypoints Estimation

This chapter presents a novel approach "Cross-Domain Hallucination Network For 3D keypoints Estimation" (CDHN) that extends the method presented in the previous chapter (Chap. 4), "Supervised Approach for 3D Keypoints Estimation From RGB Images". In this chapter, we call the previously proposed approach "Ours w.o. H-Net". Where, the H-Net represent the Hallucination network, which is the major difference between both the proposed approaches. It can be observed that the method (Ours w.o. H-Net – as shown in Fig. 4.2) uses only the RGB images to estimate 3D keypoints. Estimating 3D information directly from 2D information is a difficult task. Such approaches those only based on the images, generally could not achieve an accuracy as good as those using point cloud data as input. We observed that we could improve the accuracy of the keypoints estimation approach as presenting in the previous chapter (Fig. 4.2) by distilling the knowledge from point clouds of the original objects that are not required during the inference. Therefore as an extension, in this work we present an approach to distil knowledge from a teacher module of our network trained with 3D point cloud data and feed this information to a student module that learns to predict the features of the teacher module directly from single-view RGB images. The framework of our approach (which is inspired from [43, 27, 103, 38, 26]) is illustrated in Fig. 5.1.

In step 1, the network is trained with the teacher module (encoder *E3* fed with point cloud data) to estimate 3D keypoints from images and point clouds. In step 2, the student module (encoder *E2*) learns from the pretrained teacher module to hallucinate (i.e., produce) 3D features from RGB images. In step 3, at inference time, instead of using the teacher module,

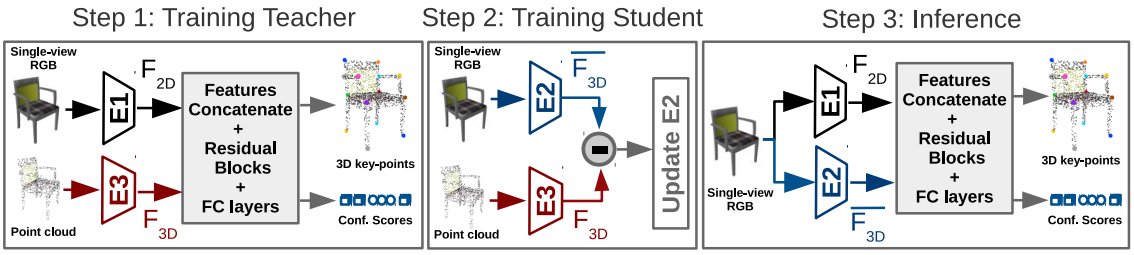


Figure 5.1 Overview of the proposed approach. In the first step, the network is trained with a teacher module (encoder *E3*) to estimate 3D keypoints from images and point clouds. In step 2, the hallucination student module (encoder *E2*) learns to produce 3D features using the pretrained teacher module. In step 3, the network uses the student module instead of the teacher in order to estimate 3D keypoints only from images. In addition, the network also predicts confidence scores to identify valid keypoints among the predicted ones.

the network uses the student module and estimates the 3D keypoints from single-view images without using the point cloud data.

Our contributions are as follows:

- We present an approach that leans to produce 3D features directly from RGB images without using point clouds
- The proposed approach estimates keypoints from single-view RGB images by leveraging information learnt from 3D data during training.
- Our approach outperforms the State-Of-The-Art (SOTA) approaches for all the categories.

5.1 Proposed approach

Given a single-view RGB image we seek to estimate an ordered list of 3D keypoints that best describes the PoI of an object. Such keypoints should be semantically and geometrically consistent for different viewing angles of an object. The proposed approach estimates total N keypoints along with their confidence scores. These scores indicate the probability of the predicted keypoints being valid for an object. So, the number of valid keypoints could vary for different objects.

The architecture of the proposed approach is illustrated in Fig. 5.2. It is based on three basic modules: feature extractor, residual blocks and 3D keypoints estimator. For extracting the features from images and point clouds, three different encoders are used. Encoders *E1* and *E2*

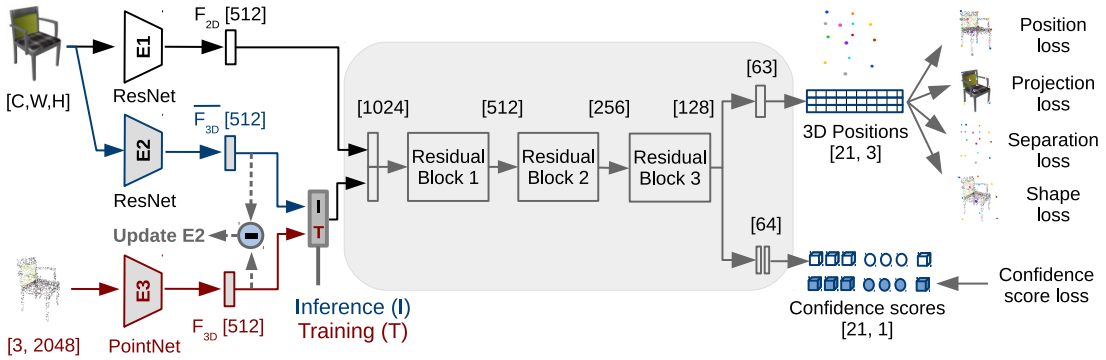


Figure 5.2 Proposed architecture – In first step, the teacher module (*E3*) that extracts 3D features (F_{3D}) from point clouds is used in the network along with the encoder *E1* that extracts 2D features (F_{2D}) from images. Both the 2D and 3D features are concatenated and are utilized in the network training for estimating the 3D keypoints. In second step, keeping the teacher modules frozen, the student module (*E2*) is trained to learn from the pretrained teacher module *E3* to produce 3D features ($\overline{F_{3D}}$) from RGB images that are similar to those of F_{3D}. In third step, during inference, the student module *E2* is used in the network that enables estimating 3D keypoints only from images. Furthermore, the network also estimates confidence scores that represent a validity of every estimated keypoint.

use the ResNet backbone [36] to extract 2D (F_{2D}) and 3D ($\overline{F_{3D}}$) features, respectively, from the given RGB image. The encoder *E3* is based on PointNet [78] and it extracts 3D features (F_{3D}) form point clouds during training. The 2D (F_{2D}) and 3D (F_{3D} or $\overline{F_{3D}}$) features are concatenated and passed to the three cascaded residual blocks. Each residual block (detailed in Fig. 5.3) contains a pair of linear layers with batch normalization connected via ReLU, and a skip connection with a single linear layer. Finally, the refined features of the residual blocks are used by keypoints estimator (having two branches based on linear layers) that estimates keypoints' position in 3D space $[x, y, z]$ and their respective confidence scores $[0\ to\ 1]$. The confidence scores reflect probabilities of the keypoints to be valid for an object. All the keypoints with the score greater than the threshold (τ) are considered as valid for the test object. We select τ as 0.5.

The network is trained in a teacher-student fashion. In the first step, the teacher module (*E3*) is trained with the network to estimate 3D keypoints from both single-view RGB images and point clouds. In the second step, the student module (*E2*) is trained to learn from the pretrained teacher module *E3* to hallucinate (i.e. produce) 3D features ($\overline{F_{3D}}$) from RGB images that are similar to those of point clouds (F_{3D}). At this time, the teacher module is frozen so only the *E2* module is updated. During inference, the pipeline uses modules *E1* and *E2* to extract the features F_{2D} and $\overline{F_{3D}}$, respectively, from a single RGB input image.

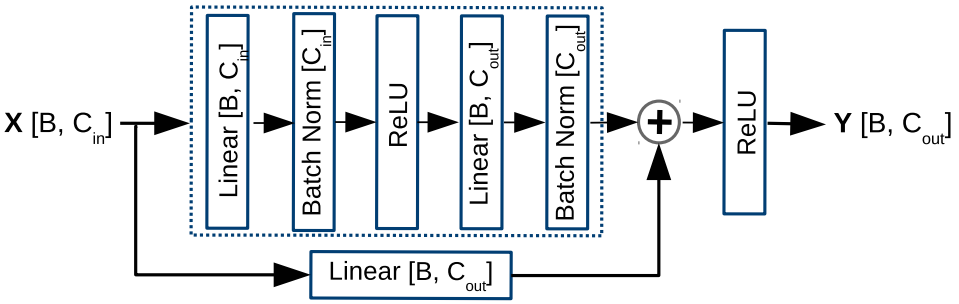


Figure 5.3 Architecture of the residual blocks. C_{in} and C_{out} are the respective lengths of the input and output features, and B denotes the batch size.

Estimation of 3D keypoints position

The 3D positions are estimated by minimizing four losses, viz., 3D position loss, 2D projection loss, separation loss and shape consistency loss. The loss functions are exactly the same as the ones used in the earlier proposed approach presented in Chap 4. However, for easy visualization, we write the functions here. For the detailed description, see Sec. 4.1 of the Chap. 4.

Considering N to be an upper bound for the number of keypoints across all the shape classes, the valid keypoints for every object could be less than or equal to N. In the ground truth, keypoints are arranged in an ordered list. Consider a set $C_Q = \{cq_1, \ldots, cq_N\}$ representing keypoint validity in the ground truth such that $cq_k = 1$ if keypoint exists at index k else $cq_k = 0$. If the object contains M valid keypoints, then we have $\sum_{k=1}^{N} cq_k = M$. We then create two sets $\mathscr{P} = \{p_k | k = 1, ..., N \; if \; cp_k == 1\}$ and $\mathscr{Q} = \{q_k | k = 1, ..., N \; if \; cp_k == 1\}$ containing the predicted and ground truth 3D positions, respectively, for the valid keypoints. Here, the cardinality of the sets is $|\mathscr{P}| = |\mathscr{Q}| = M$.

Using the above notations, the loss functions to estimate the 3D position of the keypoints can be described as follows;

– *Position loss ($\mathscr{L}_{pos}$)*: computes an error between 3D positions of the estimated and the ground truth keypoints.

$$\mathscr{L}_{pos} = \frac{1}{M} \sum_{i=1}^{M} \left\| p_i - q_i \right\|_2^2$$

– *Projection loss ($\mathscr{L}_{proj}$)*: computes an error between the 2D projected positions (pixel coordinates) of the estimated and the ground truth keypoints.

$$p_i = [x_{p_i}, y_{p_i}, z_{p_i}]^\top, \;\; q_i = [\bar{x}_{q_i}, \bar{y}_{q_i}, \bar{z}_{q_i}]^\top,$$
$$[u_i, v_i]^\top = P(p_i), \;\; [\bar{u}_i, \bar{v}_i]^\top = P(q_i),$$
$$\mathscr{L}_{proj} = \frac{1}{M} \sum_{i=1}^{M} \left\| [u_i, v_i]^\top - [\bar{u}_i, \bar{v}_i]^\top \right\|$$

– *Separation loss ($\mathscr{L}_{sep}$)*: penalizes the condition where the Euclidean distance among the keypoints estimated for a single object is less than the pre-defined hyperparameter δ^2 which is selected as 0.05.

$$\mathscr{L}_{sep} = \tfrac{1}{M^2} \sum_{i=1}^{M} \sum_{j \neq i}^{M} \max(0, \delta^2 - \left\| p_i - p_j \right\|_2^2)$$

– *Shape loss ($\mathscr{L}_{shape}$)*: computes the distance (d_i) between every keypoint from its nearest neighbour point in the ground truth point cloud. Thus, it forces the keypoints to be estimated closer to the object.

$$d_i = \left\| p_i - kNN(p_i, \mathscr{PC}) \right\|_2,$$
$$\mathscr{L}_{shape} = \frac{min(1,C)}{max(1,C)} \sum_{i=1}^{M} d_i, \quad \text{if } \; d_i > \gamma,$$

where, γ represents the tolerance threshold and is selected as 0.05.

Estimation of confidence scores

Confidence scores play a vital role in the proposed approach because it helps the network, during the inference, in identifying the valid keypoints out of the estimated N keypoints. A detailed description of the confidence scores has been presented in the previous chapter (Chap 4).

To estimate the confidence loss, instead of using the absolute loss between the prediction and the ground truth labels as we did in the previous chapter, we use the *cross entropy loss*. The notion is to consider the confidence score estimation as a classification problem that classifies the keypoints with respect to the object's shape. The *confidence score loss* ($\mathscr{L}_{conf}$) can be defined as:

$$\mathscr{L}_{conf} = -\frac{1}{N} \sum_{k=1}^{N} cq_k \cdot log(cp_k) + (1 - cq_k) \cdot log(1 - cp_k), \tag{5.1}$$

where $cp_k \in C_P = \{cp_1, \ldots, cp_N\}$ could range from 0 to 1, and $cq_k \in C_Q$ could be either 1 or 0 representing the keypoint validity in the ground truth.

The confidence and keypoints position losses are used to update the weights of the image based 2D (*E1*) and the point cloud based 3D (*E2*) encoder. So the combined loss ($\mathscr{L}_{model}$) can be defined as a weighted sum of all the above losses as:

$$\mathscr{L}_{model} = \alpha_{pos}.\mathscr{L}_{pos} + \alpha_{proj}.\mathscr{L}_{proj} + \alpha_{sep}.\mathscr{L}_{sep} + \alpha_{shape}.\mathscr{L}_{shape} + \alpha_{conf}.\mathscr{L}_{conf}. \quad (5.2)$$

In order to balance the effect of every loss $[\alpha_{pos}, \alpha_{proj}, \alpha_{sep}, \alpha_{shape}, \alpha_{conf}]$ are selected heuristically (by considering the ablations presented in Tab. 5.6) as [1, 0.33, 1, 0.5, 1], respectively.

Estimation of 3D features ($\overline{F_{3D}}$)

Since our objective is to estimate 3D keypoints from single-view RGB images, the network estimates 3D features from the same RGB images which are used to extract the 2D features. To do so, the network uses a ResNet based encoder *E2*. For the encoder's optimization, we compute the *feature loss* ($\mathscr{L}_{feature}$) by comparing the 3D features ($\overline{F_{3D}}$) computed from images (by encoder *E2*) with those (F_{3D}) computed from the point clouds (by encoder *E3*). It can be described as:

$$\mathscr{L}_{feature} = \frac{1}{K}\sum_{i=1}^{K}\left\|\overline{F_{3D}} - F_{3D}\right\|_1, \quad (5.3)$$

where K is the total number of extracted features. The loss is used to update weights of the encoder *E2*, by freezing the rest of the network modules.

Inference

During inference, the network does not require point clouds. Instead it uses only the single-view images to extract both 2D (F_{2D}) and 3D ($\overline{F_{3D}}$) features. Both the features are concatenated before forwarding to the residual blocks. The rest of the network modules (residual blocks and keypoint estimator) are the same as training network. Moreover, the network uses the confidence scores for classifying the valid keypoints instead of using ground truth information.

5.2 Experimental setup

In this section, we present implementation details, the dataset that is used in the experiments, and the performance evaluation metrics.

5.2.1 Implementation details

For image based encoders (*E1 and E2*) we use the ResNet-18 which is pretrained on the ImageNet dataset [18]. Both the encoders have same architecture, however, their weights are not shared. We update the last layers in order to get the features of dimension 512×1. For extracting the features from point clouds, we use a classification network of the PointNet (*E3*). However, instead of using the Multilayer Perceptron (MLP) after the global feature layer, we add linear layers to produce features of the same dimensions as the dimensions of the features of *E1* and *E2* (512×1). Moreover, it is trained from scratch. The proposed network is implemented in PyTorch and trained using the Adam optimizer. We perform two trainings; first, we train the encoder *E1* and *E3*, and second, we train the encoder *E2*. The learning rate in both cases is set to 10^{-5}.

5.2.2 Dataset

We use the KeypointNet [130] dataset in our experiments. Moreover, we use our extended version of the KeypointNet dataset with 24 random rotations. We load the data images, point clouds, and the corresponding ground truth keypoints simultaneously in every batch. The images and point clouds are passed to the proposed network in pairs to estimate the keypoints in 3D space. The ground truth keypoints are used to evaluate the estimates. We use the same data splits as provided by KeypointNet dataset. We evaluate the proposed approach for the public categories of the KeypointNet dataset. However, for comparison with the KP-Net and StarMap, we consider the same three categories as considered by KP-Net.

5.2.3 Performance metrics

We test the proposed network for two views (A and B) of the same object. The network estimates two sets of valid keypoints, one for each view. We use Procrustes analysis [82] to estimate transformation (E_T) between both sets of the predicted keypoints. We evaluate the performance of our approach by considering the same two metrics as we used in the previous chapter (Chap. 4). Using the first metric, we compute the angular distance error (E_T) between the estimated (R_{est}) and the ground truth (R_{gt}) transformation matrices as:

$$E_T = 2\, arcsin\left(\frac{1}{2\sqrt{2}}\, ||R_{est} - R_{gt}||_F\right).$$

Using the second metric, we compute the angular distance error between the 3D positions ($\mathscr{A}_{est} = \{ae_i | i = 1, ..., M\}$) of the valid estimated keypoint and the corresponding positions ($\mathscr{A}_{gt} = \{ag_i | i = 1, ..., M\}$) of the ground truth keypoints as:

$$E_P = \frac{1}{M}\sum_{i=1}^{M} arccos\left(\frac{\mathbf{ae}_i \,.\, \mathbf{ag}_i}{|\mathbf{ae}_i|\,|\mathbf{ag}_i|}\right).$$

For a fair comparison with the KP-Net, we consider the first evaluation. Nevertheless for validation on other categories, results from both evaluations are presented.

5.3 Results and comparison

Comparison with baseline approaches

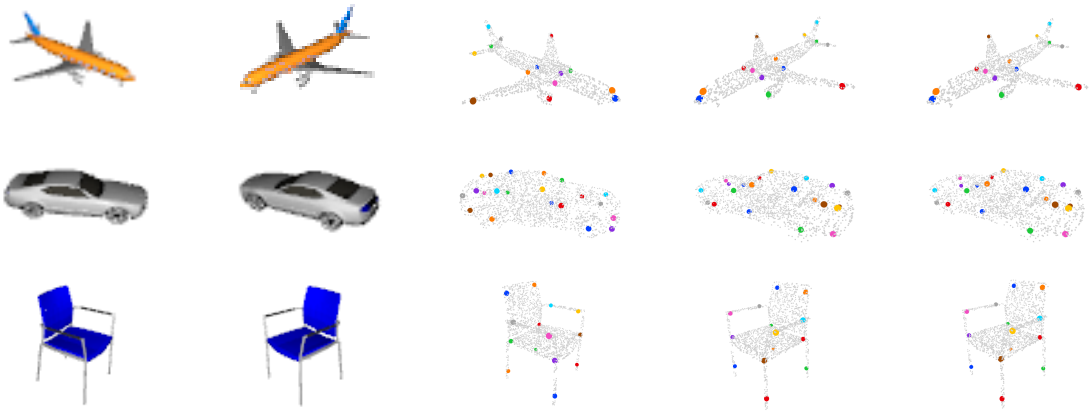
We compare against KP-Net [98], StarNet [145] and the previously proposed approach in the Chap. 4 as being the methods that compute 3D keypoints from a single-view RGB image at a testing time as we do. Since, the major difference between our previous and the newly proposed approach is the Hallucination Network (H-Net), in this chapter we call the previously proposed approach is as "Ours w.o. H-Net".

By following the same procedure as reported in the KP-Net paper [98], we load the test set in pairs of images, representing two random views of the same object. The network estimates the keypoints for both views which are later used for computing the relative pose. The error in estimated and ground truth poses is calculated using E_T as done by KP-Net. We consider the results of KP-Net as reported in [98], whereas, we use the publicly available StarNet's model that is trained on Pascal3D+ dataset [118]. For a fair comparison, we test on the same synthetic test set (in the next sections, we also present its performance for a realistic test set). Furthermore, before computing the relative pose we normalize the 3D keypoints estimated by the StarMap with respect to the corresponding ground truth point clouds. The comparison is given in Tab. 5.1.

Angular distance errors between the keypoints estimated in two views by the proposed CDHN are always (for all the categories) lower than those of the SOTA approaches, i.e. four versions of the KP-Net, StarMap and our approach without H-Net. The performance of the CDHN is superior because, unlike our method, the 3D keypoints estimation module of the KP-Net and StarMap rely on 2D key points, so their 3D estimation may contain errors [49]. Secondly, they do not use 3D data during training for reasoning in the 3D space (i.e., finding the 3D positions). In comparison, our method leverages point cloud data for estimating a sparse set of 3D keypoints and learns to generate 3D features directly from single-view images. Interestingly, the error is relatively high for chair category in all the approaches. This happens because of large intra-class shape variation for this category, yet the angular distance error for CDHN is lower than that of SOTA approaches.

Table 5.1 Comparison with the SOTA approaches based on E_T. Mean and median angular distance errors are calculated using the ground truth and the estimated rotations between two views of a same object.

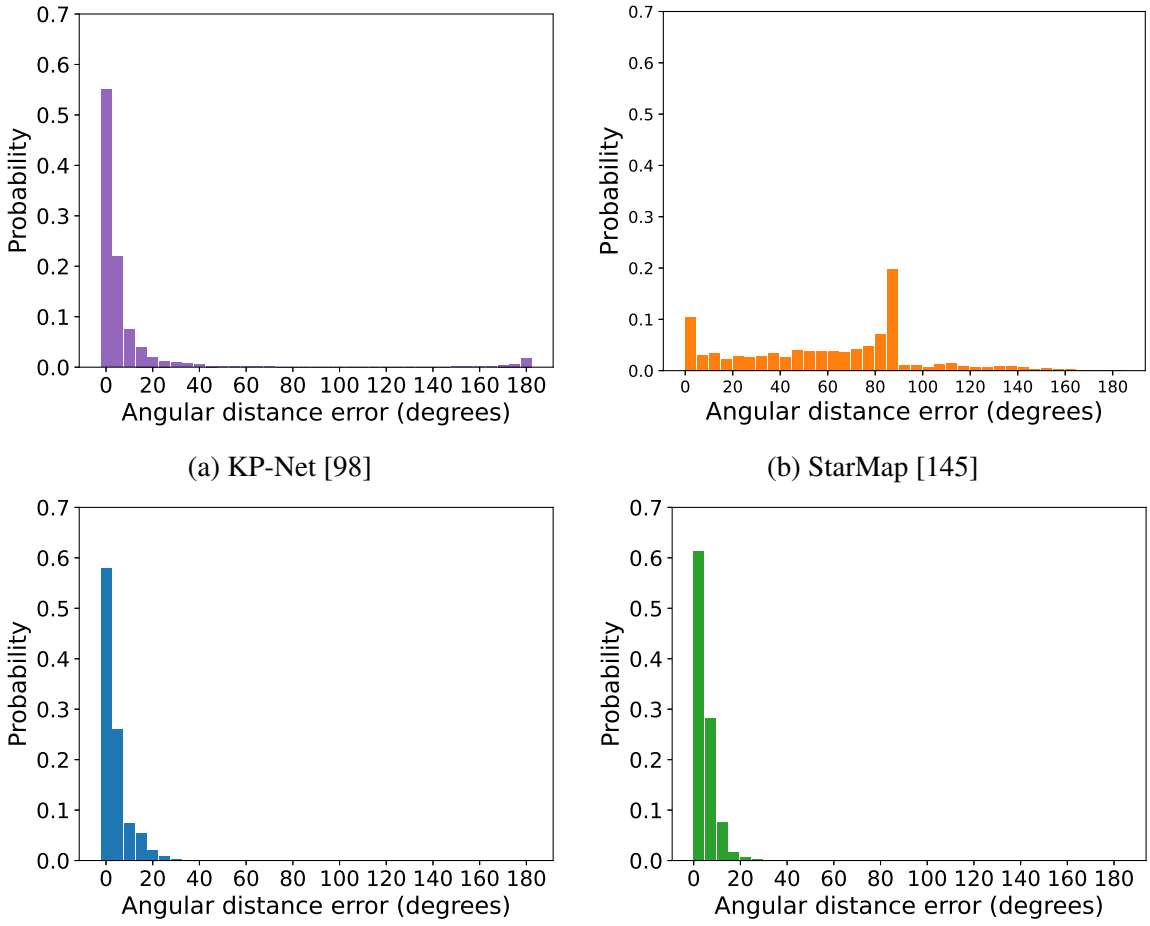
Method	Airplane ↓		Car ↓		Chair ↓	
	Mean	Median	Mean	Median	Mean	Median
Sup. KP-Net	18.350	7.168	16.268	5.583	21.882	8.771
Sup. KP-Net + O-Net	17.800	6.802	13.961	4.475	20.502	8.261
KP-Net + O-Net	18.561	6.407	13.500	4.418	14.238	5.607
KP-Net	17.330	5.721	11.310	3.372	14.572	5.420
StarMap	57.89	64.04	64.31	67.93	61.39	69.01
Ours w.o. H-Net	3.257	2.053	5.190	2.073	10.732	4.096
Ours (CDHN)	**3.171**	**2.048**	**5.057**	**2.057**	**9.582**	**4.084**



(a) View A (b) View B (c) KP A (d) KP B (e) R_{est}(KP A)

Figure 5.4 Visualizations of the keypoints estimated by CDHN, for computing a pose between two views (a) and (b) of an object. The corresponding estimated keypoints are shown on the original point clouds in (c) and (d). The keypoints of view A (c) are transformed to view B using the estimated rotation matrix as illustrated in (e). It can be seen that the keypoints in (d) and (e) lie in very similar places. Also, their semantic order is maintained.

The qualitative results of CDHN are given in Fig. 5.4, showing the predicted keypoints (Fig. 5.4c, Fig. 5.4d) for two views A and B (Fig. 5.4a, Fig. 5.4b) of the same object. R_{est} denotes the estimated transformations between views A and B. The transformed version of the predicted keypoints for view A using R_{est} are shown in Fig. 5.4e. It can be observed that the predicted keypoints for view B (Fig. 5.4d) and the transformed version of view A (Fig. 5.4e) look very similar. This indicates that the estimated pose is almost the same as the ground truth pose.



(a) KP-Net [98]

(b) StarMap [145]

(c) Ours w.o. H-Net

(d) Ours (CDHN)

Figure 5.5 Distribution of angular distance error between ground truth and predicted relative rotations for two random views of an object. The error distribution is averaged across car, airplane, and chair categories.

We also present a comparison (in Fig. 5.5) based on the distribution of the angular distance errors between ground truth pose and the pose estimated by the SOTA approaches between the two views of an object. It shows that the maximum pose error of CDHN lies between 0° to 20° which is lower than those of SOTA approaches. Moreover, 61.2% of the pose error lies between 0° to 5° in case of the CDHN which is approximately 6.2% and 3.2% more than those of KP-Net (55%) and our approach without H-Net (58%), respectively. Furthermore, the errors also lie from 170° to 180° in the case of the KP-Net which is not the case in our approach. The errors of the StarMap lie between 0° to 150°, which shows that the estimated keypoints are not more useful for relative pose estimation. For more clarity, we also present the Cumulative Distribution Function (CDF) of the average angular distance errors in Fig. 5.6.

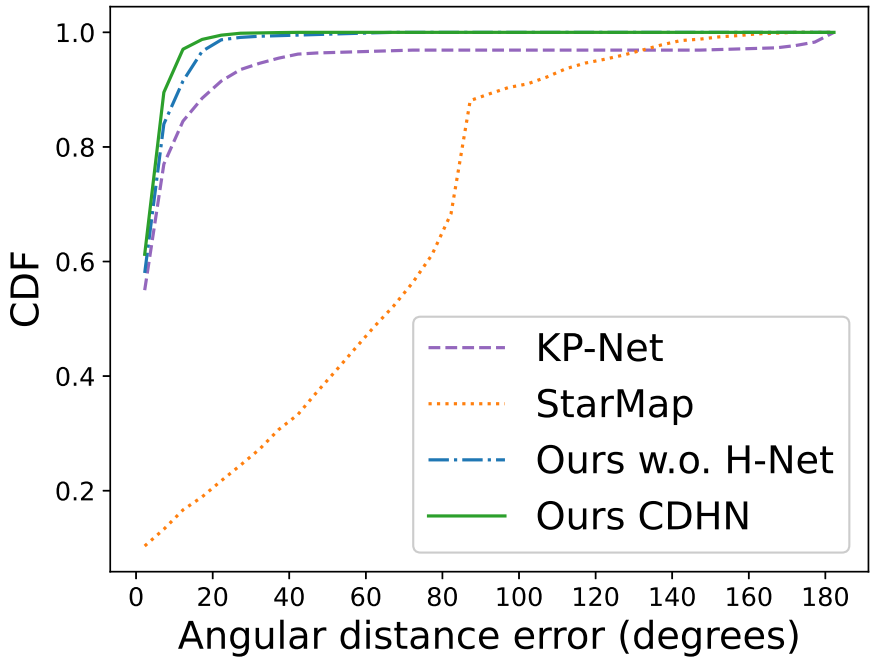


Figure 5.6 Cumulative Distribution Function (CDF) of the average angular distance errors depicted in the Fig. 5.5.

It can be observed that the CDF of the CDHN (continuous green plot) is comparatively higher than the other approaches (i.e., CDHN reaches the maximum CDF value before the other approaches).

A conclusion from this experimental evaluation is that the proposed CDHN outperforms the existing SOTA approaches as well as the previously proposed approach without H-Net.

Evaluations for other categories

The proposed approaches (with and without H-Net) are evaluated for the other KeypointNet categories. In this experiment, results are evaluated using both the performance metrics E_T and E_P as given in Tab. 5.2. The proposed approaches remain successful in estimating keypoints for other categories. The lower angular distance error shows a correspondence between the keypoints predicted in the two views. It can be observed that the errors for all the categories are lower in case of CDHN. The error is slightly high for some categories due to different reasons including the structural variation (single/bunk beds, tables), different keypoints for similar object shapes (helmet, knife, etc.), and differences in the center of rotation and the center of mass of the object (i.e. mug) and symmetry of the shapes. Qualitative results are given in Fig. 5.7. Rows 1 and 4 represent the images used for the evaluation. Rows

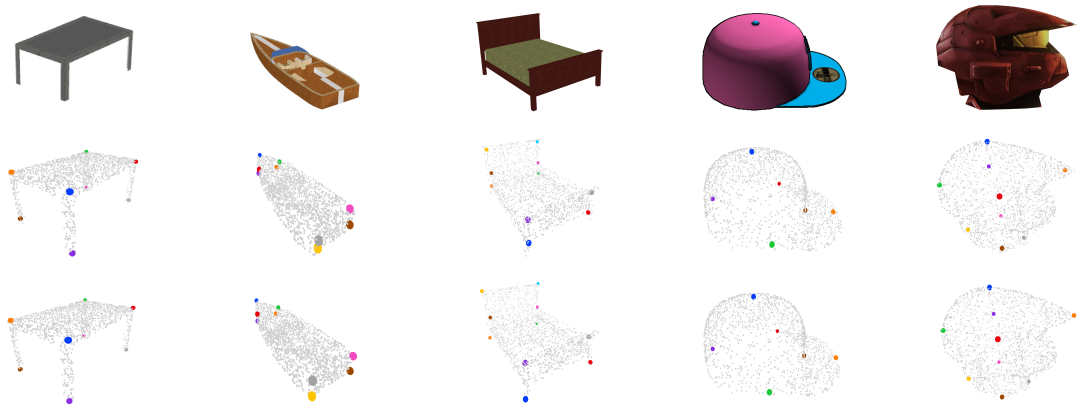
Table 5.2 Performance evaluation of the proposed approaches (with and without H-Net) for the other categories. Angular distance error in the pose estimated between two views is calculated using the both performance metrics (E_T and E_P).

Category	Ours w.o. H-Net [148]				Ours (CDHN)			
	$E_T \downarrow$		$E_P \downarrow$		$E_T \downarrow$		$E_P \downarrow$	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
Table	23.92	3.63	18.08	2.97	20.32	3.36	16.03	2.82
Vessel	14.65	4.39	11.66	3.48	11.80	4.18	9.29	3.36
Bed	28.60	12.42	25.33	9.05	27.32	8.36	19.03	7.82
Cap	16.90	8.19	13.63	6.26	15.52	8.01	12.32	6.19
Helmet	26.95	16.06	23.50	15.24	23.37	14.23	17.74	10.36
Knife	25.33	13.01	20.60	12.49	22.76	12.62	18.19	11.43
Motorcycle	9.47	3.23	6.49	2.51	9.01	3.05	5.08	2.28
Guitar	19.56	5.29	7.25	2.93	18.13	4.85	6.76	2.87
Mug	18.47	9.14	10.32	5.94	13.01	7.07	9.63	4.68
Bottle	17.12	14.85	14.67	12.01	15.84	13.19	13.72	11.03
Average	20.10	9.02	15.15	7.29	17.71	7.89	12.78	6.28

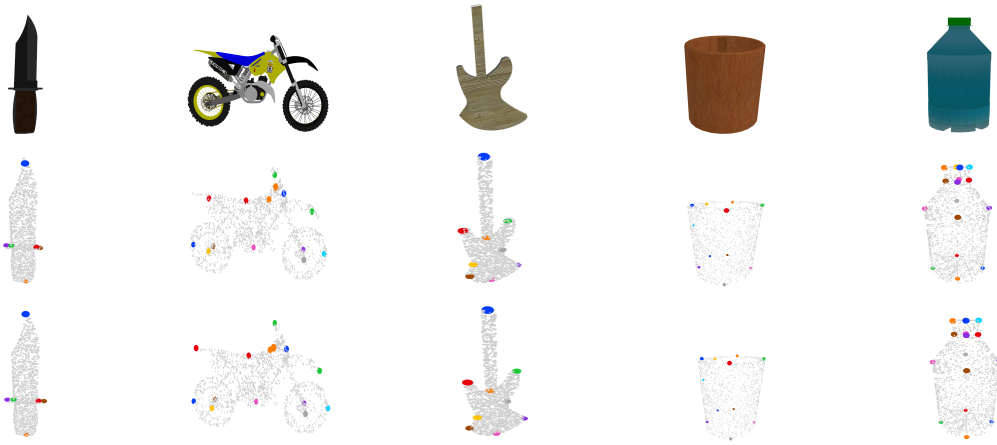
2 and 5 show the keypoints estimated by the proposed approach, and their corresponding ground truth keypoints are illustrated in rows 3 and 6. These results indicate that the proposed method successfully estimates the 3D keypoints, not just for the airplane, car and chair categories, but also for other categories having varying and complex geometries.

Comparison on the realistic dataset

To test our approach on more realistic images (close to real images), we render the images by placing the object in the center of the real backgrounds selected randomly from the SUN dataset [119]. The rest of the evaluation procedure is the same as discussed earlier. Since KP-Net [98] has not reported quantitative results for real/realistic dataset in their paper, we compare our approach with the StarMap. The comparison is given in Tab. 5.3. This can be observed that the angular distance error of StarMap is almost the same for both the datasets; synthetic and realistic. It is due to two reasons; first, the 3D keypoints are estimated using the 2D keypoints, which are the same for both datasets, second, the 3D keypoints are estimated for non-occluded regions and thus are not feasible for relative pose estimation. Therefore, the error is comparatively high for all the evaluated categories. In comparison, CDHN outperforms the SOTA approaches including the previously proposed network without H-Net. This validates the significance of the H-Net in the presented CDHN.



(a) Table (b) Vessel (c) Bed (d) Cap (e) Helmet



(f) Knife (g) Motorcycle (h) Guitar (i) Mug (j) Bottle

Figure 5.7 Qualitative results of the proposed approach for remaining categories. Row (1, 4) show the input images, row (2, 5) and row (3, 6) present the corresponding estimated and ground truth keypoints, respectively. It can be visualized that the proposed approach estimates a semantically ordered list of keypoints even for the occluded parts of the objects.

5.4 Ablation studies

5.4.1 Significance of the confidence scores in our approach

Since, the confidence scores help in classifying the valid and invalid keypoints from the predicted N keypoints, here we evaluate their classification accuracy. To do so, first we classify the estimated keypoints using the confidence scores and the ground truth information. Then the similarity in the classified valid and invalid keypoints is computed, that represents a classification accuracy of the confidence scores. Since, the confidence score is based on the

Table 5.3 Evaluation of the proposed approach on the realistic dataset. The relative angular distance error between two views of an object has been improved by the proposed CDHN.

Method	Airplane ↓		Car ↓		Chair ↓	
	Mean	Median	Mean	Median	Mean	Median
StarMap	60.17	64.34	64.42	69.70	76.11	85.81
Our w.o. H-Net	51.010	29.270	41.470	12.840	70.782	61.520
Our (CDHN)	37.324	14.302	29.147	8.638	59.322	46.849

threshold τ, we repeat the experiment for its different values. The classification accuracy (in %) is given in the Tab. 5.4. It can be noticed that the accuracy of the valid selections is above 93% for $\tau \leq 0.5$.

Table 5.4 Classification accuracy (in %) of the valid estimated keypoints by the confidence scores w.r.t. those using the ground truth information. The results for different values of τ are presented.

τ	0.99	0.90	0.80	0.70	0.60	0.50	0.40	0.30	0.20	0.10
Airplane	91.99	92.34	92.67	92.88	93.02	93.15	93.25	93.36	93.44	93.56
Car	96.76	96.92	97.07	97.15	97.22	97.26	97.31	97.37	97.42	97.47
Chair	96.29	96.33	96.32	96.29	96.25	96.17	96.10	96.02	95.88	95.68

Furthermore, a bar chart illustrating the count of the estimated valid keypoints is shown in the Fig. 5.8. It can be observed that the total classified valid keypoints are approximately equal in both the cases; either we use predicted confidence scores or the ground truths for valid keypoints selections. We identified that the 95.53% of the keypoints are correctly classified for $\tau \geq 0.5$. The percentage slightly increases for $\tau < 0.5$, however, it may consider a wrong keypoint with probability less than 0.5.

5.4.2 Estimated ($\tau \geq 0.5$)s. ground truth valid keypoints

Furthermore, we show the error in estimated pose between the two views of an object using the valid predicted keypoints selected for $\tau \geq 0.5$ and the keypoints identified using ground truth information. This comparison is shown in Tab. 5.5 for both the performance metrics (E_T and E_P). The Standard Error (SE) is calculated as $\sigma/\sqrt{n}$, where σ is the standard deviation of n angular distance errors. From Tab. 5.5, we observed that the two sets of keypoints lead to similar results.

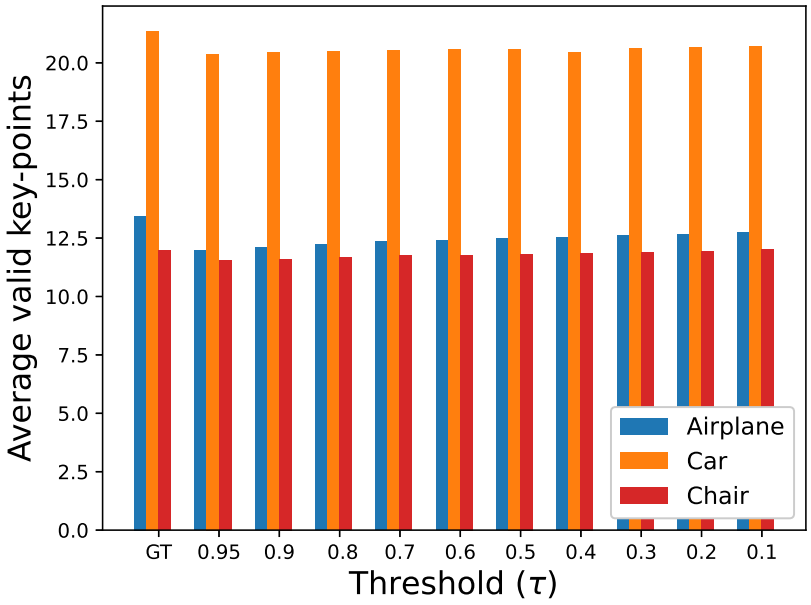


Figure 5.8 Average valid keypoints per category. Comparison of the valid keypoints selected based on confidence scores for different τ with the ground truth keypoints (leftmost).

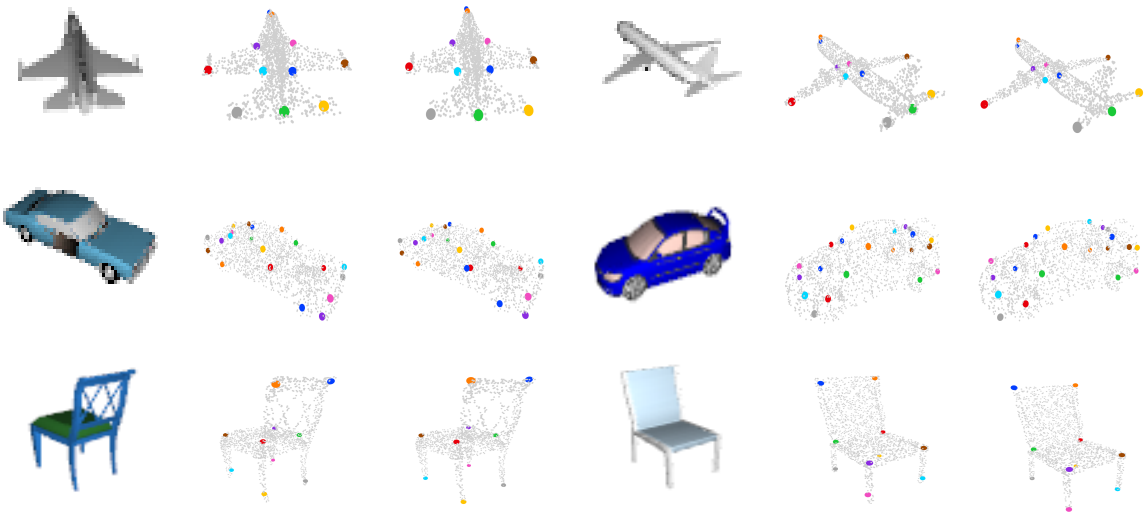
5.4.3 Performance for selected losses.

In order to identify the contribution of every loss in the proposed approach, we train our network by omitting each loss one by one from the five keypoints estimation losses. The results as given in Tab. 5.6 highlight that the network with all the losses performs overall well. Moreover, the position and the confidence score loss significantly influence the model's performance. It is because the prediction of an accurate 3D position and the identification of valid keypoints are the essential elements for 3D keypoints estimation. Shape loss plays a vital role by forcing the keypoints towards the object's surface, where as the contribution of the projection and the separation loss is comparatively low. These losses are more important for the approaches that estimate 2D keypoints first and then use them to estimate depth information. The qualitative results for every experiment are illustrated in Figs. 5.9 to 5.13. In every figure, columns (a) and (d) show two different test images; the predicted keypoints are illustrated in (b) and (e), and (c) and (f) are the corresponding ground truth keypoints, respectively.

The performance of the network without $\mathscr{L}_{sep}$ (Fig. 5.9) and $\mathscr{L}_{proj}$ (Fig. 5.10) is slightly decreased in comparison with the network with all the losses. The results of the network trained without $\mathscr{L}_{shape}$ are illustrated in Fig. 5.11. It can be observed that some of the

Table 5.5 Comparison of the valid estimated keypoints selected by the confidence scores (Conf.) for $\tau \geq 0.5$ with those selected using the ground truth information (GT). Mean and Standard Error (SE) of the angular distance error between two views of an object is computed using both the evaluation metrics.

Category	$E_T \downarrow$				$E_P \downarrow$			
	Mean		SE		Mean		SE	
	Conf.	GT	Conf.	GT	Conf.	GT	Conf.	GT
Airplane	3.171	3.169	0.074	0.072	2.816	2.794	0.001	0.001
Car	5.058	5.057	0.238	0.215	3.893	3.859	0.005	0.004
Chair	9.597	9.582	0.325	0.310	7.287	7.273	0.007	0.004



(a) Input (b) Prediction (c) GT (d) Input (e) Prediction (f) GT

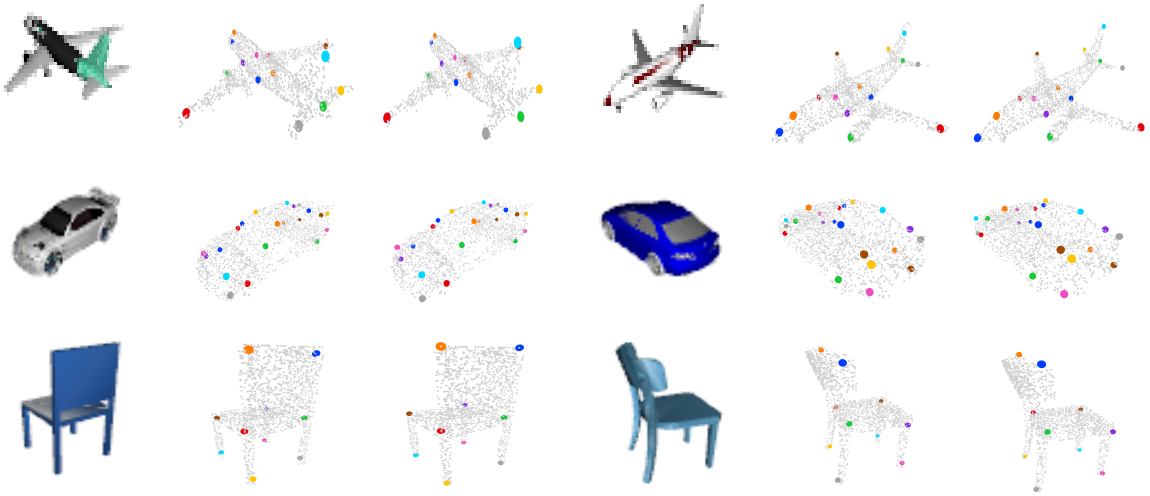
Figure 5.9 Qualitative results of the proposed network trained without $\mathscr{L}_{sep}$.

keypoints are predicted outside the object (in the surrounding). this is because we have ignored the loss ($\mathscr{L}_{shape}$) that minimizes the distances of the estimated keypoints from the surface of the object.

Fig. 5.12 shows the keypoints estimated by the network that is trained without the confidence loss ($\mathscr{L}_{conf}$). The network predicts random confidence which could be correct or incorrect. We select the keypoints for which the confidence scores are greater than or equal to 0.5. It can be observed that the number of predicted valid keypoints are less than those in the ground truth. Moreover, sometimes invalid keypoints are also classified as valid due to the incorrect confidence scores. Due to these erroneous keypoints, the error in pose estimation is increased.

Table 5.6 Performance of the proposed approach (CDHN) for selected losses.

Loss	Airplane ↓		Car ↓		Chair ↓		Avg. Err. inc.↓	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
				E_T				
All losses	3.17	2.05	5.06	2.06	9.58	4.08	–	–
w.o. $\mathscr{L}_{sep}$	3.18	2.05	5.06	2.06	9.59	4.09	0.01	0.01
w.o. $\mathscr{L}_{proj}$	3.19	2.09	5.09	2.07	9.94	4.09	0.14	0.02
w.o. $\mathscr{L}_{shape}$	4.39	2.88	7.45	2.57	12.84	4.92	2.29	0.73
w.o. $\mathscr{L}_{conf}$	40.39	19.75	32.36	10.18	35.95	14.80	30.29	12.18
w.o. $\mathscr{L}_{pos}$	79.10	72.96	94.59	99.54	75.26	65.90	77.05	76.73
				E_P				
All losses	2.79	2.03	3.86	2.06	7.28	4.07	–	–
w.o. $\mathscr{L}_{sep}$	2.80	2.03	3.86	2.06	7.28	4.07	0.01	0.01
w.o. $\mathscr{L}_{proj}$	2.83	2.04	3.89	2.07	7.45	4.08	0.08	0.01
w.o. $\mathscr{L}_{shape}$	3.94	2.47	6.10	2.38	9.60	4.64	1.90	0.45
w.o. $\mathscr{L}_{conf}$	26.39	12.76	19.74	8.48	20.54	10.15	17.58	7.75
w.o. $\mathscr{L}_{pos}$	56.00	54.18	68.44	70.34	56.58	47.91	55.70	54.76



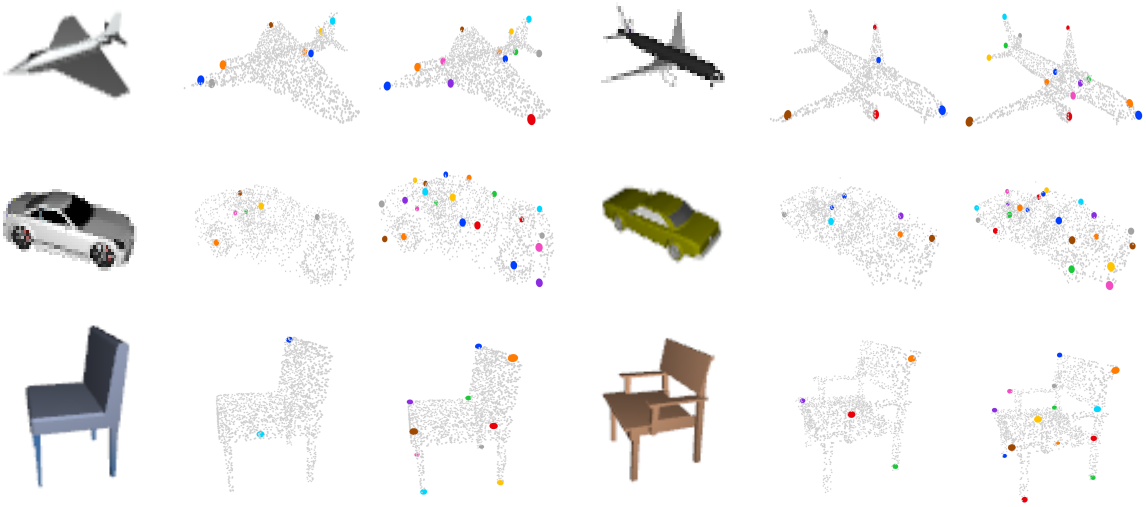
(a) Input (b) Prediction (c) GT (d) Input (e) Prediction (f) GT

Figure 5.10 Qualitative results of the proposed network trained without $\mathscr{L}_{proj}$.

Fig. 5.13 shows the visualizations of the keypoints estimated by the network that is trained without the position loss ($\mathscr{L}_{pos}$). All the estimated keypoints are correctly identified and these lie on the surface of the objects. However, they are in random 3D positions, i.e., their order is not maintained. These experiments highlight that confidence and position loss can have a significant influence on performance.

(a) Input (b) Prediction (c) GT (d) Input (e) Prediction (f) GT

Figure 5.11 Qualitative results of the proposed network trained without $\mathscr{L}_{shape}$.

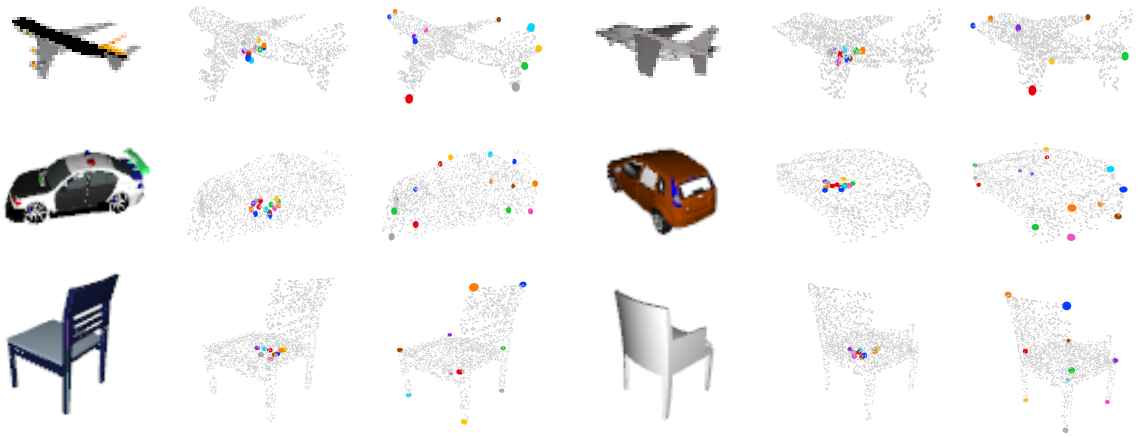


(a) Input (b) Prediction (c) GT (d) Input (e) Prediction (f) GT

Figure 5.12 Qualitative results of the proposed network trained without $\mathscr{L}_{conf}$.

5.4.4 Performance of the Hallucinated module.

This ablation highlights the performance of the hallucinated 3D features from the student (*E2*) module of the network. The proposed approach is evaluated separately for both the teacher (*E3*) and the student module. It is found that the error in pose estimation is low when the teacher module is used. However, in the case of the student module, the error has increased by 1.09° and 0.9° on scale of E_T and E_P, respectively (compared in Tab. 5.7). This



(a) Input (b) Prediction (c) GT (d) Input (e) Prediction (f) GT

Figure 5.13 Qualitative results of the proposed network trained without $\mathscr{L}_{pos}$.

Table 5.7 Performance of the approach for the teacher and the student module.

Metric	Module	Airplane ↓		Car ↓		Chair ↓	
		Mean	Median	Mean	Median	Mean	Median
E_T	Teacher (*E3*)	2.847	2.016	4.062	2.019	7.628	4.039
	Student (*E2*)	3.171	2.048	5.057	2.057	9.582	4.084
E_P	Teacher (*E3*)	2.212	2.013	3.233	2.014	5.773	3.340
	Student (*E2*)	2.794	2.024	3.859	2.055	7.273	4.068

result is due to two reasons: first, the features extracted by the teacher module from the point clouds (F_{3D}) are more accurate than those ($\overline{F_{3D}}$) extracted by the student module from the images, and second, the student module can learn up to a limited level. Nevertheless the angular distance error of the network using the student module (5.936°) is 8.46° lower than that of KP-Net (14.40°).

5.5 Chapter summary

The chapter presents an approach that estimates 3D keypoints from single-view RGB images. During training, it exploits 3D features extracted from the object's point clouds to learn to produce similar 3D features from RGB images. In inference, the network extracts both the 2D and 3D features from images without requiring the object's point clouds. The estimated keypoints are compared with those of SOTA approaches by utilizing them for finding the pose between two views of an object. Moreover, unlike the existing approaches, the proposed

approach computes confidence scores for every predicted keypoint. The scores allow the network to classify valid keypoints from the total N predicted keypoints. Thus, the approach is not limited to a fixed number of keypoints, it can predict different keypoints based on the object's shape. Furthermore, this characteristic of the proposed approach allows it to train jointly for several categories.

The presented results validate that the estimated keypoints can be used to compute the relative pose between objects. Moreover, they can also be used to estimate an object's pose, especially in 3D reconstruction tasks [17, 147, 12], which estimate an object's shape without pose information.

Although the approach presented in this chapter outperforms the existing image based approaches, it is a supervised approach and hence requires a huge dataset containing human-annotated ground truth keypoints. Creating such datasets and annotating all the objects, especially in real scenes is a resource-consuming task. Due to this limitation, very limited datasets have been created so far [89]. Moreover, supervised approaches may not be considered as generalized solutions as they are applicable to a fixed list of categories depending upon the dataset.

Consider an unsupervised approach as a solution to the above-mentioned problems of the supervised methods, in the next chapter, we present a novel architecture that estimates the 3D keypoints from PCDs which out requiring ground truth keypoints.

Chapter 6

SC3K: Self-supervised and Coherent 3D Keypoints Estimation from Point Clouds

This chapter presents an approach to estimate 3D keypoints on the surface of an object in a self-supervised way. Unlike the previously proposed approaches, this approach does not require human-annotated keypoints to train the network. Therefore the estimated keypoints are different in nature then those estimated by the supervised approaches. The supervised approach estimates the keypoints closer to the ground truth keypoints which are selected by a person, whereas, the presented self-supervised approach estimates the keypoints that best characterize the object's shape. This is achieved by combining different loss functions that force the keypoints to cover the complete object and appear closer to its surface. This chapter also addresses problems associated with the supervised keypoints estimation methods and the presents un-/self-supervised approaches as a solution.

The literature reports that representing 3D objects using a set of keypoints [11, 48, 101] is a common and fundamental step for several geometrical reasoning tasks, including shape registration, object tracking, pose estimation, action recognition, shape deformation, retrieval and reconstruction [89, 129, 108, 143, 41]. Because extracting such keypoints is the first processing step, it is crucial that keypoints are extracted reliably from Point Cloud Data (PCD) of object shapes, as an error might affect negatively any further high-level tasks.

The solution to this problem was initially cast as a supervised learning task: given a dataset of manually annotated PCDs with keypoints, a computational model infers the keypoints position given a PCD as input [113, 148, 60, 40, 130]. While these methods provided impressive results on the dataset they were trained on, they also made clear the limitations of

supervised approaches. The basic issue is the requirement of having large enough datasets containing well defined ground truth annotations for every object. Annotating such datasets is hard, finding keypoints in 3D requires a long time from the user, noise or missing data on the PCD can compromise quality, and highly symmetric/smooth objects might confuse the annotator in finding the correct keypoints.

Considering such limitations, recent methods have focused on not-supervised approaches to bypass the need for human annotations. Self-supervision methods define proxy tasks for which a large number of annotations can be obtained during training [111, 77, 129, 10, 133], e.g. geometrical transformations, canonical mapping, reconstruction to learn the prototype of intra-class object, etc. [84, 141, 83, 95, 100]. Unsupervised approaches differently promote keypoints that are implicitly given by reasoning on the object geometry, e.g. point-level clustering, object's skeleton, consistency between object's symmetry, part contrasting, etc. [67, 121, 92, 41, 131].

The shift to these learning paradigms clearly allows to generalise keypoint extraction but not without drawbacks. No having human annotations means that the exact identification of a specific keypoint in a particular semantic 3D region is not guaranteed when intra-class variations are present (see Fig. 6.1) and especially when some shape elements might be missing or repeated (e.g. a table can have 4 or more legs). Moreover, for several applications such as shape registration, it is paramount to maintain the semantic consistency of keypoints, i.e., their vector ordering, as an output of the network architecture. Intra-class variations might also bias the network to localise keypoints where no PCD is present simply because in some shape instances we have a semantic region there (e.g. an additional leg of a table in some object samples). In addition to all these considerations, keypoints extraction has to be robust against common perturbations of PCDs, semantic ordering and the accuracy in localising the keypoints should be preserved even if PCDs are rotated, noisy and sub-sampled (decimated) as shown in Fig. 6.1.

To this end, we propose a self-supervised training strategy and model architecture to train a keypoints detector with such desired properties. Our training strategy feeds pairwise randomly rotated versions of the same object PCD for each sample in the class. For each rotated shape, the network estimates independently a set of 3D keypoints. This initial keypoints estimation backbone optimises a loss promoting keypoints that are not-overlapping, close to the original PCD and covering the volume of the whole shape. These two sets of keypoints are then refined in two steps. First, we transform both the sets in their known canonical pose and then we compute one-to-one consistency between the corresponding

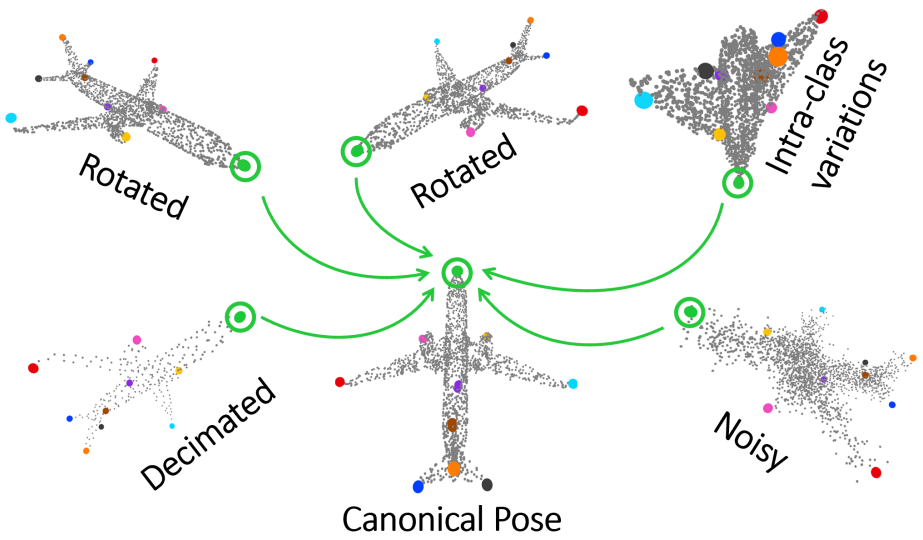


Figure 6.1 Self-supervised and unsupervised keypoints estimation from point cloud data has to be robust to perturbations such as rotations, intra-class shape variations, noisy data and an arbitrary number of input 3D points. The keypoint localisation has not only to be accurate and pertain to the object surface but it should also preserve semantic coherence, as shown in this figure by the green keypoint which is always associated with a specific object region despite arbitrary variations in the point cloud.

keypoints of the sets. Introducing a component of the loss function that penalises incorrect matches allows the estimation of consistent keypoints irrespective of the pose of the input PCD. Second, to refine the position and semantic coherence of the estimated keypoints, we compute the relative pose between the two sets of keypoints as a proxy task and we then minimise the error against the known relative pose of the PCDs pair. Such training strategy and network architecture promote the inference of keypoints that are semantically coherent, robust to perturbations, and with high accuracy.

To summarise, the main contributions of this work are as follows:

- The proposed network estimates 3D keypoints from a single PCD in a generic pose as opposed to the other methods in the State-Of-The-Art (SOTA) which require a PCD in a canonical pose.
- The presented two-step learning procedure allows to estimate keypoints that are semantically consistent for intra-class objects regardless of perturbations, such as rotation, noise, or down-sampling;
- On average, the presented approach outperforms the SOTA approaches and it is able to generalise to novel object poses.

6.1 Proposed approach - SC3K

This section describes every module of the proposed network and provides details of the loss functions used during training and the experimental setup.

6.1.1 Proposed Architecture

Given a PCD of an object, the goal of the proposed approach, named SC3K, is to estimate keypoints that are semantically coherent and accurate despite arbitrarily rotated PCDs and perturbations, without requiring ground annotations. The architecture of the SC3K is illustrated in Fig. 6.2.

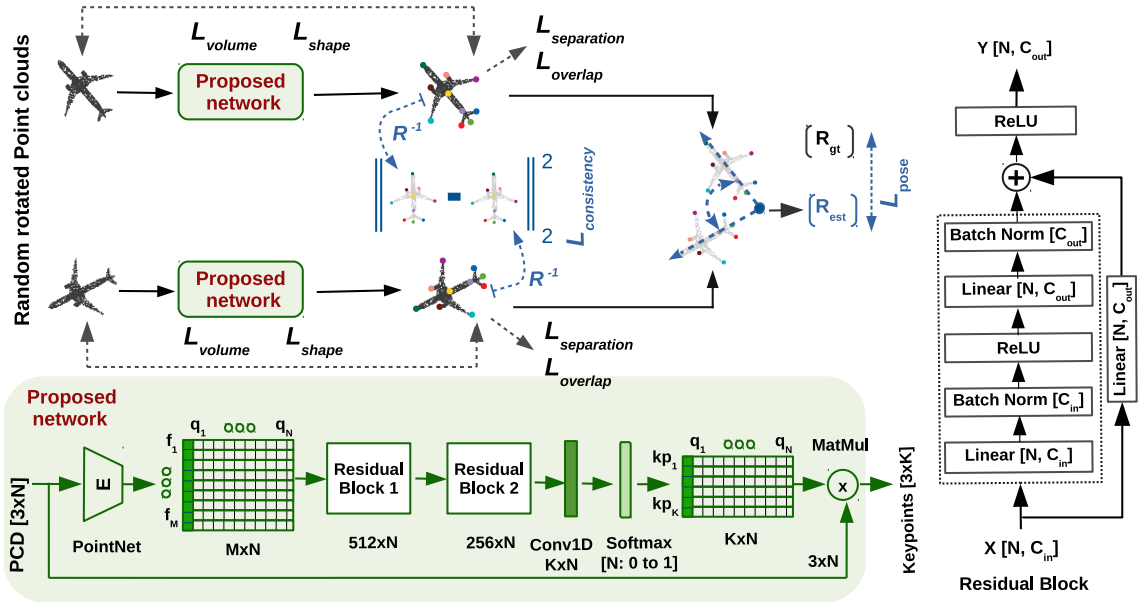


Figure 6.2 Network architecture – The proposed network takes a PCD of N points as input and extracts M global features for every point using PointNet encoder. The features are passed by two cascaded residual blocks followed by a convolutional and a softmax layer in order to estimate $K \times N$ features. Where K is the number of keypoints and N defines the weights of the points in the input PCD to be selected as keypoints. Finally, K 3D keypoints are computed as weighted average points of the input PCD. To make the estimated keypoints pose coherent and semantically consistent, we first estimate keypoints for two randomly rotated versions of the PCD and then compute a mutual loss between keypoints in two steps (as highlighted in navy blue). First, both the keypoints sets are transformed to the canonical pose and are used to compute one-to-one consistency between the corresponding keypoints. Second, the relative pose between the two keypoints sets is compared with those of the original PCDs. The proposed network is illustrated in lower part of the figure and the residual block 1 and 2 are the same as shown in the right part.

The approach uses a PointNet [78] backbone to extract M features for every point in the input PCD (Proposed Network in Fig. 6.2). The extracted features pass through two consecutive residual blocks that reduce the features from M to 256. Each residual block (right scheme in Fig. 6.2) contains a pair of linear layers with batch normalisation connected via ReLU, and a skip connection with a single linear layer. The refined features are later projected to a conv1D and a softmax layer to estimate $K \times N$ probabilities, where K represents the total number of keypoints and N (probabilities) represents the weight for every point in the input point cloud to be selected as the keypoints. The weights of every keypoint ($N \times 1$) are multiplied to the original PCD ($3 \times N$) in order to estimate the final keypoint (3×1). The final keypoint represents a weighted average point of the PCD. We repeat this process K times to estimate all the keypoints ($3 \times K$).

6.1.2 Training procedure

The proposed training procedure accepts as input an object PCD that is then randomly rotated twice to obtain two PCDs. These PCDs are then processed by the proposed network that outputs two sets of keypoints. This pairwise set will be used as a self-supervised signal to enforce keypoints semantic consistency. For each set of keypoints, a loss with four components is computed, based on how well the keypoints fit the single shape of the original PCD. We call this loss "position loss". Then, the two sets of keypoints from the two randomly rotated PCDs are used to compute "mutual dependency loss" in two steps. In the first step, both the keypoints sets are transformed to the (known) canonical pose to compute the one-to-one consistency between the corresponding keypoints. In the second step, the relative pose of the keypoints are compared with those of the input PCDs to refine the keypoints position and the semantic coherence. The network is trained to minimise both loss functions. In the following, we will present these two losses in detail.

6.1.3 Position loss

The proposed network takes a single shape $\mathscr{P} = [p_1, p_2, ..., p_N] \in \mathbb{R}^{3\times N}$ as input and outputs a set of K keypoints $\mathscr{K} = \{k_1, k_2, ..., k_K\}, \in \mathbb{R}^{3\times K}$ with $K \ll N$. We use the original $\mathscr{P}$ and the estimated $\mathscr{K}$ to compute the position loss. The desired keypoints properties are that keypoints should not overlap with each other, be relatively separated and cover as much as possible the whole object volume while still being close to the PCD. We will describe in detail these four loss components in the next paragraphs.

Overlap loss: To avoid multiple keypoints being estimated at the same 3D position we define the overlap loss as:

$$\begin{aligned} \mathscr{L}_{overlap} &= \frac{1}{K^2}\sum_{i=1}^{K}\sum_{j=1}^{K}\left[\left\|k_i - k_j\right\|_2 < \tau_1\right], \quad i \neq j \\ \left[\left\|k_i - k_j\right\|_2 < \tau_1\right] &= \begin{cases} 1 & \text{if true} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{6.1}$$

where [.] is the Iverson bracket. The loss $\mathscr{L}_{overlap}$ counts the total number of overlapping keypoints. Two keypoints are considered as overlapping if the Euclidean distance between them is less than the threshold τ_1, which is 0.05.

Separation loss:

This loss ($\mathscr{L}_{sep}$) maximises the distance of every keypoint (k_i) from its k-nearest neighbour keypoints ($kNN(k_i, \mathscr{K})$) in $\mathscr{K}$ thus promoting more spread out configurations of points. The loss is defined as:

$$\mathscr{L}_{sep} = \frac{1}{max(\frac{1}{K}\sum_{i=1}^{K}\left\|k_i - kNN(k_i, \mathscr{K})\right\|_2, 0.01)}, \tag{6.2}$$

where, 0.01 in the denominator is to avoid division by zero.

Shape loss: Since $\mathscr{L}_{sep}$ moves away keypoints from their neighbours without any maximum distance limit, keypoints might move easily far from the object PCD and even further. Therefore, we use the shape loss ($\mathscr{L}_{shape}$) that enforces keypoints being closer to the object's shape. The loss minimises the distance of every keypoint k_i in $\mathscr{K}$ from its nearest neighbour point in the original $\mathscr{P}$. The loss can be defined as:

$$\mathscr{L}_{shape} = \frac{1}{K}\sum_{i=1}^{K}\left\|k_i - kNN(k_i, \mathscr{P})\right\|_2. \tag{6.3}$$

Volume loss: The $\mathscr{L}_{sep}$ and $\mathscr{L}_{shape}$ losses do not consider how the keypoints are distributed over the whole shape of the object. Therefore to estimate keypoints that cover the entire object, we compute the volume loss as $\mathscr{L}_{volume}$. The loss computes the difference between the longest diagonal of the 3D bounding box of the estimated keypoints with that of the

original PCD as:

$$\mathscr{L}_{volume} = smoothL_1(vol(\mathscr{K}) - vol(\mathscr{P})), \tag{6.4}$$

where $vol()$ is the function that accepts a set of points ($\mathscr{K}$ or $\mathscr{P}$), identifies a maximum and a minimum point from the accepted set, and returns their difference (i.e., the longest diagonal distance of the object's bounding box). We consider this diagonal distance as approximate volume by following [24]. To find the difference in volume, we use smooth *L1* loss as this loss is less sensitive to outliers compared to the MSE loss [30].

The total position loss can be summarised as a weighted sum of the above four loss components;

$$\mathscr{L}_{position} = w_{sep} \cdot \mathscr{L}_{sep} + w_{ovr} \cdot \mathscr{L}_{overlap} + w_{sh} \cdot \mathscr{L}_{shape} + w_{vol} \cdot \mathscr{L}_{volume}, \tag{6.5}$$

where, $\{w_{sep}, w_{ovr}, w_{sh}, w_{vol}\}$ are not optimised hyperparameters fixed to $\{0.05, 0.05, 4, 1\}$ respectively.

6.1.4 Mutual dependency loss

In order to refine the positions of the keypoints and to make them semantically coherent across different rotations of an object, we use the mutual dependency loss. Differently from the positional loss, here we consider the pair of keypoints obtained from the randomly rotated shapes. The loss is given by two components as described below.

In detail, suppose that the two randomly rotated versions of the input PCDs are $\mathscr{P}_{\mathscr{A}} = [a_1, a_2, ..., a_N] \in \mathbb{R}^{3\times N}$ and $\mathscr{P}_{\mathscr{B}} = [b_1, b_2, ..., b_N] \in \mathbb{R}^{3\times N}$ while the K keypoints estimated by the proposed approach for each input PCD can be represented as $\mathscr{K}_{\mathscr{A}} = [k_1^a, k_2^a, ..., k_K^a], \in \mathbb{R}^{3\times K}$ and $\mathscr{K}_{\mathscr{B}} = [k_1^b, k_2^b, ..., k_K^b], \in \mathbb{R}^{3\times K}$, respectively. Then the loss functions can be described as given below.

Step 1 – Keypoints consistency loss:

Consider that $R_a \in \mathbb{R}^{3\times 3}$ and $R_b \in \mathbb{R}^{3\times 3}$ are the rotations associated to $\mathscr{P}_{\mathscr{A}}$ and $\mathscr{P}_{\mathscr{B}}$, respectively. We use these rotation matrices and transform the keypoints ($\mathscr{K}_{\mathscr{A}}$ and $\mathscr{K}_{\mathscr{B}}$) back to their canonical pose. The keypoints are said to be coherent if they overlap in this common reference system and if their indexes exactly match. To introduce this desiderata, we compute the consistency loss ($\mathscr{L}_{consist}$) between the corresponding keypoints in both the

transformed sets as:

$$\mathscr{L}_{consist} = \frac{1}{K}\sum_{i=1}^{K} \left\| R_a^{-1} k_i^a - R_b^{-1} k_i^b \right\|_2^2. \tag{6.6}$$

In this way, to push the keypoints with the same indexed to be in the same 3D location, we penalise the keypoints with the wrong ordering and 3D position.

Step 2 – Pose loss: In order to emphasise even more the estimation of coherent keypoints that are less sensitive to the object's pose, we train the network to solve an auxiliary and self-supervised keypoints registration task, by estimating the rotation matrix that aligns the two sets against the (known) ground truth. Suppose R_{est} is the relative pose between the estimated keypoints $\mathscr{K}_{\mathscr{A}}$ and $\mathscr{K}_{\mathscr{B}}$, computed by using orthogonal Procrustes Analysis. Then the pose loss ($\mathscr{L}_{pose}$) can be computed using the Frobenius norm between the R_{est} and relative pose of the PCDs ($R_{ba} = R_a \cdot R_b^T$) as:

$$\mathscr{L}_{pose} = 2\, arcsin\left(\frac{1}{2\sqrt{2}}\, ||R_{est} - R_{ba}||_F\right). \tag{6.7}$$

It can be observed that if the keypoints in the canonical pose are not aligned/overlapped, the R_{est} will be erroneous, and hence the loss will be high. In other words, the lower pose loss validates the accuracy of the correspondences in the two sets of keypoints.

The mutual dependency loss can be defined as the weighted sum of the above two losses:

$$\mathscr{L}_{mutual\_dependency} = w_{con} \cdot \mathscr{L}_{consist} + w_{pose} \cdot \mathscr{L}_{pose}, \tag{6.8}$$

where $\{w_{con}, w_{pose}\}$ are defined as $\{1, 0.05\}$. The overall training loss is the sum of the position and the mutual dependency loss;

$$\mathscr{L}_{overall} = \mathscr{L}_{position} + \mathscr{L}_{mutual\_dependency}. \tag{6.9}$$

Inference

During inference, the proposed approach takes only an arbitrarily rotated PCD as input and it estimates a semantically ordered list of K keypoints in the same pose as the pose of the input PCD.

6.2 Experimental setup

This section presents implementation details, the dataset that is used in the experiments, performance evaluation metrics with their significance, and a comparison between our approach and the SOTA approaches.

6.2.1 Implementation details

The network is implemented in PyTorch and trained using the Adam optimizer with a learning rate of $1e^{-3}$. We do not freeze any part of the network. In all the experiments, the batch size is set to 32 and trained on a 12GB GPU. We train UCLS [24], SM [89] and our network for 200 epochs and evaluate them using the best trained model (with the minimum validation loss).

6.2.2 Dataset

We use KeypointNet dataset [130] in our experiments. It contains 8329 objects (3D model and PCDs of 16 object categories) for a total of 83231 keypoints. We do not use the ground truth keypoints. Whereas, we rotate every object in 24 random poses since during training we need to feed two rotated versions of the same object to the proposed network. We use the same rotation matrices that are used in ONet [68] with a validation and testing split that differs from the training set. For a fair comparison, we use the original (not-rotated) dataset to evaluate our and the SOTA approaches.

6.2.3 Metrics for unsupervised keypoints estimation

To compare the performance of the proposed approach, we first define the two standard metrics: inclusivity and coverage [24]. The **inclusivity metric** [24] computes the percentage of the keypoints ($\mathscr{K}$), which are estimated close to points of the input $\mathscr{P}$. A keypoint k_i whose distance d_i to the nearest neighbour point in $\mathscr{P}$ is below the predefined threshold τ_2 is considered as a close keypoint. The metric is defined as:

$$\begin{aligned} d_i &= \left\| k_i - kNN(k_i, \mathscr{P}) \right\|_2 \\ Inclusivity &= 100 \times \frac{1}{K} \sum_{i=1}^{K} \left[d_i < \tau_2 \right], \end{aligned} \tag{6.10}$$

where [.] is the Iverson bracket (as described in Eq. 6.1). Although the inclusivity loss computes how close the $\mathscr{K}$ are estimated from the original $\mathscr{P}$, it does not evaluate how well the keypoints cover the whole object. Therefore, evaluation is further supported by the **coverage metric** [24], which compares the distance between the longest diagonals of the 3D bounding boxes of the $\mathscr{K}$ with that of the $\mathscr{P}$. Assuming that the $\mathscr{K}$ and $\mathscr{P}$ are normalized, the metric can be defined as:

$$\begin{aligned} Cov &= 100 \times \left[1 - \frac{|vol(\mathscr{P}) - vol(\mathscr{K})|}{vol(\mathscr{P})}\right] \\ Coverage &= \begin{cases} Cov & \text{if } vol(\mathscr{K}) \leq 2 \times vol(\mathscr{P}) \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{6.11}$$

where $vol()$ is the same as used in Eq. 6.4. The coverage will be 100% if both bounding boxes fully overlap and it will decrease if the bounding box of the $\mathscr{K}$ is either smaller or greater than the one of $\mathscr{P}$.

6.3 Results and analysis

We compare our results with the SOTA approaches UCLS [24] and SM [89] that estimate the 3D keypoints in an unsupervised way. We trained and tested the UCLS and SM using KeypointNet [130] dataset (keeping the PCDs in canonical pose). However, considering the nature of our approach, we train it for rotated PCDs. We test our approach under two conditions: *SC3K_rot* (the PCDs with the random rotation) and *SC3K_can* (i.e PCDs in the canonical pose). The random rotations are used to evaluate the accuracy of our approach irrespective of the object's pose. However, to be consistent with our competitors (UCLS and SM), we also test our method for the original PCDs in a canonical pose.

Tab. 6.1 presents a comparison among UCLS, SM and our approach (*SC3K_can* and *SC3K_rot*) based on the two performance metrics as discussed in Sec. 6.2.3. Higher values correspond to better performance for every metric. The first inclusivity metric shows that, on average, the proposed approach (*SC3K_rot*) outperforms the SOTA approaches by estimating the keypoints close to the object's surface. However, *SC3K_can* achieves results better than those of UCLS and comparable to those of SM. The metric depends on the total number of keypoints and the tolerance threshold τ_2. To validate this, we train our network separately for different numbers of keypoints, and calculate the inclusivity for different τ_2. It is found that inclusivity is higher for fewer keypoints and it increases with the increase of the

Table 6.1 Performance comparison between the proposed approach and SOTA approaches (UCLS [24] and SM [89]) based on KeypointNet dataset. We test our approach for PCDs in canonical pose (*SC3K_can*) and the PCDs rotated in random poses (*SC3K_rot*). The results are calculated for 10 keypoints and the threshold τ_2 for the inclusivity is selected as 0.1. For all the metrics, higher values are best. The comparison validates that on average the proposed approach outperforms the SOTA approaches for all the metrics.

Category	**Inclusivity**				**Coverage**			
	UCLS	SM	SC3K_*can*	SC3K_*rot*	UCLS	SM	SC3K_*can*	SC3K_*rot*
Airplane	71.02	72.05	**87.20**	74.30	88.63	92.59	**96.34**	94.37
Bed	67.00	71.89	**80.00**	72.29	94.17	84.28	**98.20**	92.85
Bottle	75.44	72.84	77.36	**84.01**	80.93	91.44	**97.95**	94.16
Cap	57.50	59.50	56.25	**67.14**	60.83	85.01	**94.64**	91.81
Car	71.32	71.95	**76.05**	74.45	83.69	**90.69**	89.84	90.19
Chair	68.54	69.67	56.65	**72.33**	83.92	85.87	**95.31**	90.22
Guitar	50.14	69.29	**96.47**	69.04	79.83	85.65	**97.64**	92.17
Helmet	64.10	72.41	55.00	**74.68**	79.87	82.09	**90.50**	90.44
Knife	52.05	92.03	**98.33**	93.15	76.84	77.39	**98.77**	88.77
Motorbike	78.43	**95.28**	85.00	87.74	78.87	86.12	**94.34**	91.33
Mug	47.42	65.87	46.25	**82.37**	89.63	83.15	**95.15**	91.22
Table	60.06	79.13	**79.15**	73.05	82.97	91.31	**97.40**	92.32
Vessel	76.89	94.24	92.90	**95.24**	78.79	85.28	**97.18**	90.03
Average	64.61	75.86	75.89	**78.44**	81.46	86.22	**95.63**	91.53

τ_2. Fig. 6.3 shows the average inclusivity (of the test set) for different values of τ_2. We select τ_2 as 0.10 and consider 10 keypoints for the experiments and comparison. The coverage metric shows that on average the proposed approach is successful in estimating the keypoints whose 3D bounding boxes best overlap those of the original PCDs. For all the categories, *SC3K_can* achieves better results.

A qualitative comparison between the keypoints estimated by SC3K and the SOTA approaches is depicted in Fig. 6.4. For better understanding, the estimated keypoints (in different colours) are shown on top of the original PCDs (in Gray). The colour of the keypoints represents their semantic ID information, i.e. a point with the same colour should stay in the same area despite perturbations. Columns 1 and 2 illustrate the keypoints estimated by UCLS [24] and SM [89], respectively. In contrast, two views of the keypoints estimated by the proposed approach are depicted in columns 3 and 4. The comparison validates that our keypoints are estimated close to the surface, highlighting the corners, thus best characterizing the object's shape.

To evaluate the semantic consistency between the keypoints estimated for different objects of a same category, we compute the Dual Alignment Score (**DAS**) metric [89]. By following

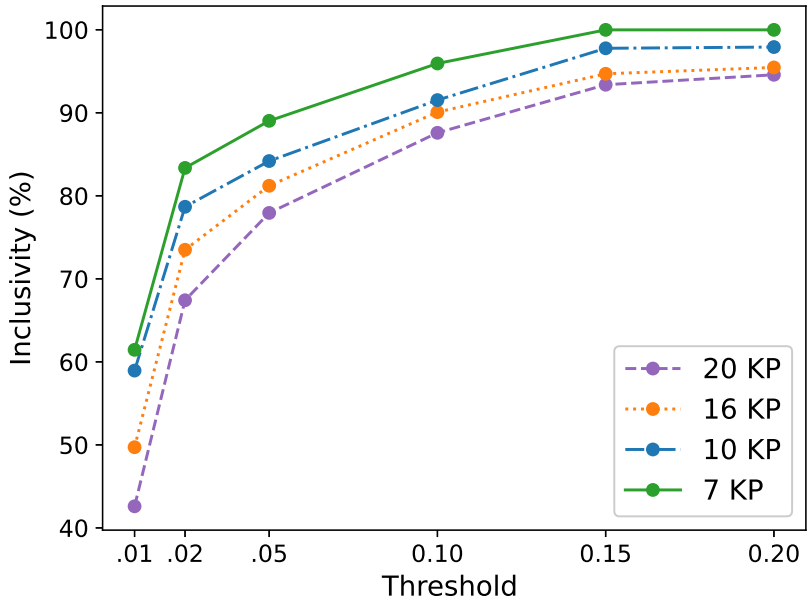


Figure 6.3 Average inclusivity of the proposed approach for different keypoints and threshold values (τ_2). The inclusivity increases with an increase in the τ_2, and it is higher for fewer keypoints.

the same procedure, we define the ratio of a set of reference keypoints for each category that are semantically aligned w.r.t. the corresponding human annotated keypoints. To compare our approach with [131], we consider the results reported in the paper, since the code is not shared publicly. The comparison is depicted in Tab. 6.2. On average, our approach (SC3K) outperforms the other approaches.

Unlike the existing approaches, we also evaluate the coherence property of the keypoints by computing the Matching Error (ME). This error is a localisation error of the keypoints given PCD perturbations. We first estimate keypoints for different rotated versions of the same object PCDs and transform them to the canonical pose using the known rotations. Since the estimated keypoints are with the correct order, we compute order-wise position error between the corresponding keypoints on the canonical reference frame. A low error would indicate that 2D position of a keypoint is rather unaffected by variations of the PCD. We repeat this procedure for all the instances of a category and calculate the ME in terms of mean error (μ) and the standard deviation (σ). The quantitative results are depicted in Tab. 6.3.

The qualitative results of this experiment are illustrated in Fig. 6.5. Columns 1 and 2 (on the left side) show the keypoints estimated for two transformed versions of the same objects.

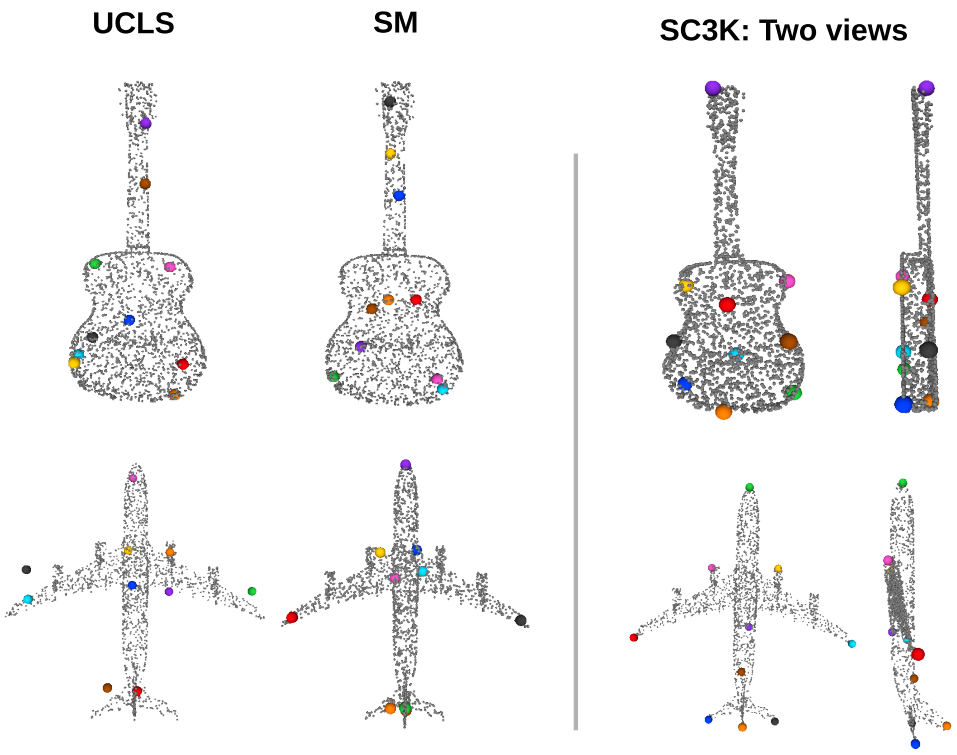


Figure 6.4 Qualitative comparison. Columns 1 and 2 present keypoints estimated by UCLS and SM, respectively. Columns 3 and 4 show the keypoints estimated by SC3K. It can be observed that some of keypoints of the UCLS are estimated outside the object (airplane). The keypoints estimated by SC3K best characterize the object's shape, as they are estimated on the surface and cover the complete object.

Table 6.2 Comparison based on the semantic consistency between the keypoints estimated for different objects of the same category. The baseline results are the same as reported in [131]. The higher value is best.

Category	UCLS [24]	SM [89]	ISS [144]	MR [131]	SC3K
Airplane	61.40	77.70	13.10	81.00	**82.86**
Chair	64.30	76.80	10.70	83.10	**87.04**
Car	–	**79.40**	8.00	74.00	75.19
Table	–	70.00	16.20	**78.50**	76.03
Guitar	–	63.10	8.70	61.30	**65.67**
Mug	–	67.20	11.20	68.20	**89.35**
Cap	–	53.00	13.10	57.10	**59.72**
Mean	62.85	69.60	11.57	71.89	**76.55**

Table 6.3 Pose coherent test: The keypoints estimated for randomly rotated versions of the same object are first transformed to the canonical pose. Then ME (in terms of μ and σ) is computed between the corresponding keypoints of all the keypoints sets. The error is very small considering that the maximum error could be $\sqrt{3}$.

ME	Airplane	Bed	Bottle	Cap	Car	Chair	Guitar	Helmet	Knife	Bike	Mug	Table	Vessel	Mean
μ	0.041	0.072	0.058	0.057	0.061	0.045	0.047	0.071	0.055	0.072	0.039	0.072	0.040	0.056
σ	0.019	0.057	0.056	0.038	0.042	0.021	0.020	0.052	0.034	0.040	0.023	0.051	0.031	0.037

It can be seen that the corresponding keypoints are semantically consistent irrespective of the object's pose, this validates the keypoints are coherent. The keypoints on the right side of the same Fig. 6.5 (columns 3, 4 and 5) illustrate the keypoints estimated for different rotated objects of the same category. It can be observed that the keypoints also maintain the correspondences across the different intra-class variations of the object class.

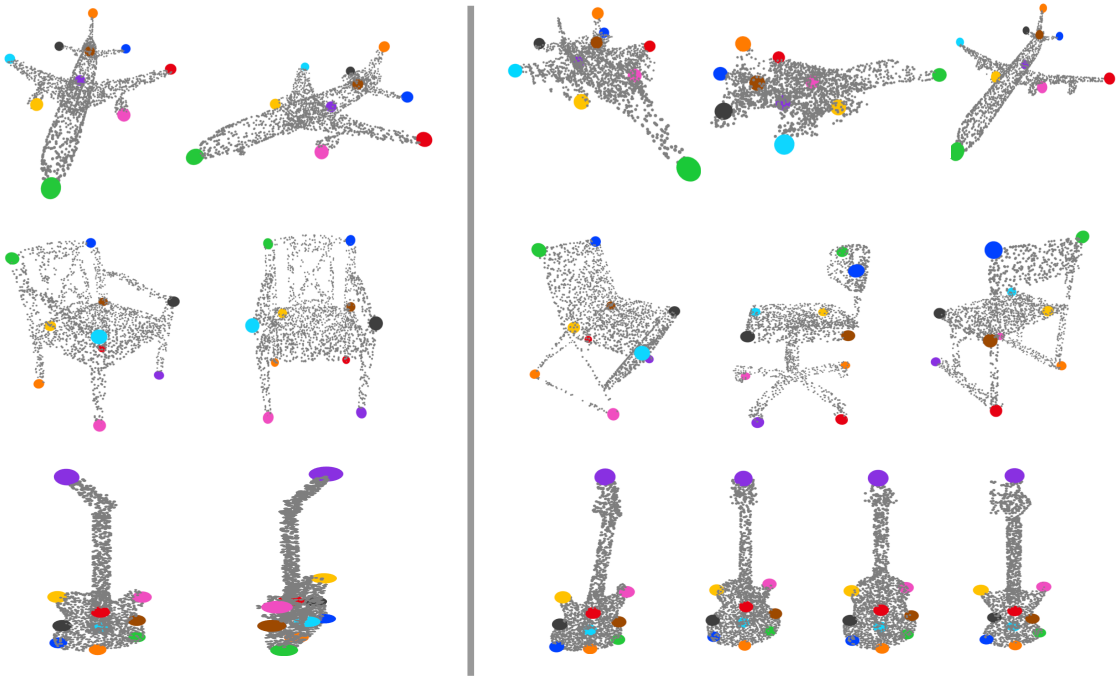


Figure 6.5 Shape pose variations and semantic correspondence: columns 1,2) keypoints estimated for two rotated versions of the same object are pose coherent; columns 3-5) keypoints semantically correspond to intra-class variations – they correspond to those estimated for different objects of the same category.

6.4 Ablation studies

This section presents three ablations: *i)* effect of the number of keypoints computed and their effect on the metrics; *ii)* evaluation and performance of the network with combinations of the different training losses; *iii)* effect of varying noise ratio and decimations of the PCDs.

6.4.1 Effect of the number of keypoints

We evaluate our approach by varying the number of computed keypoints from the PCD. We found that for most of the shapes (e.g. bottle, guitar), our approach estimates keypoints over the surface of the object. However, for the detailed objects with gap between the parts (i.e., airplanes have relevant empty spaces between a wing and the tail), some of the keypoints are estimated outside the object (in the gaps). This effect appears only when a high number of keypoints are considered (higher than 35). As an example, different number of keypoints estimated by our approach for the cup and airplane category are shown in Fig. 6.6.

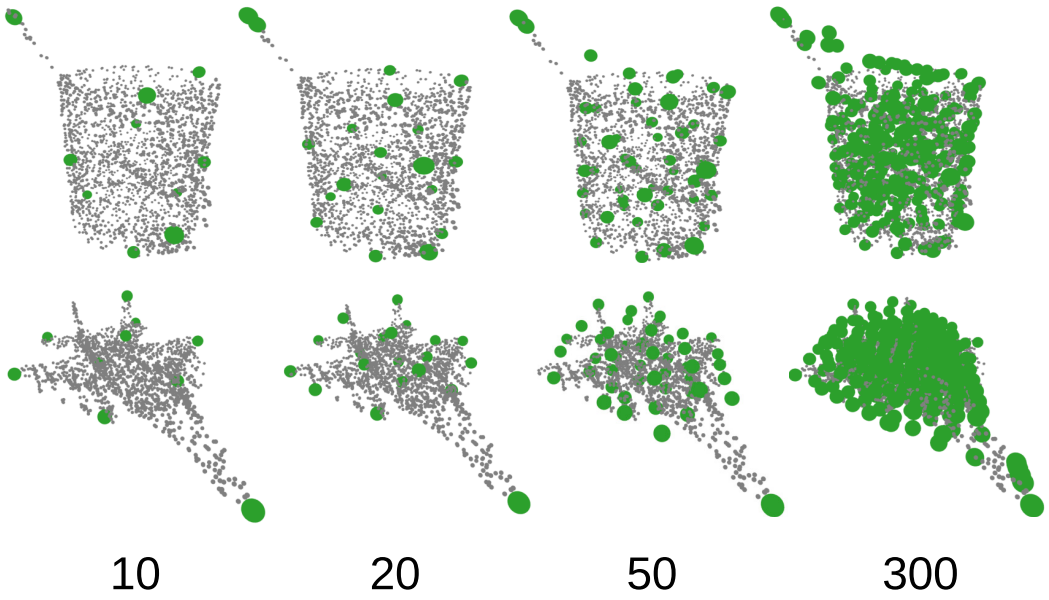


Figure 6.6 Estimation of different number of keypoints for the same object. The keypoints are estimated on the object's surface if they are less than or equal to 35 in number. They are predicted outside the object (in case of more than 35 keypoints), especially for the detailed objects having empty spaces among the object's parts.

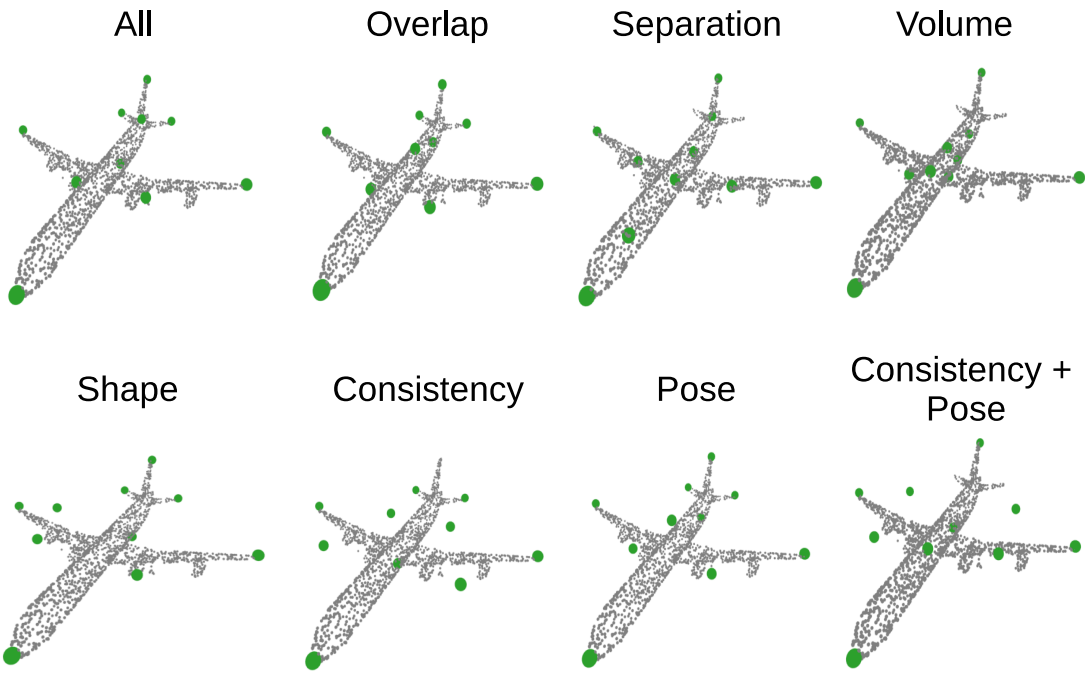


Figure 6.7 Performance of our approach with different combinations of losses. The leftmost figure shows the keypoints when the network is trained for all the losses. In the remaining figures, the model is trained without a specific loss which is mentioned at the top of every figure.

6.4.2 Performance for the selected losses

In order to highlight the significance of every loss in the proposed approach, we train and evaluate the network by ignoring each loss one by one. The results are illustrated in the Tab. 6.4. The conditional formatting green-to-red shows high-to-low values. The table shows that the network performs overall well when all the loss functions are used. The overlap loss contributes comparatively low and is required only at the beginning of the training when the keypoints are estimated randomly. The contribution of the separation loss is comparatively higher than the overlap, shape and volume loss since it maintains the distance between the estimated keypoints, thus enforcing the keypoints to move over the whole object and toward the surface. Shape loss avoids the estimation of the keypoints outside the object. The contribution of the volume loss is comparatively lower than the other loss functions. The consistency and pose losses allow the estimation of the corresponding and pose coherent keypoints. Ignoring both losses at the same time affects the overall performance of the proposed approach. The qualitative results of the proposed approach trained without the selected loss function are illustrated in Fig. 6.7.

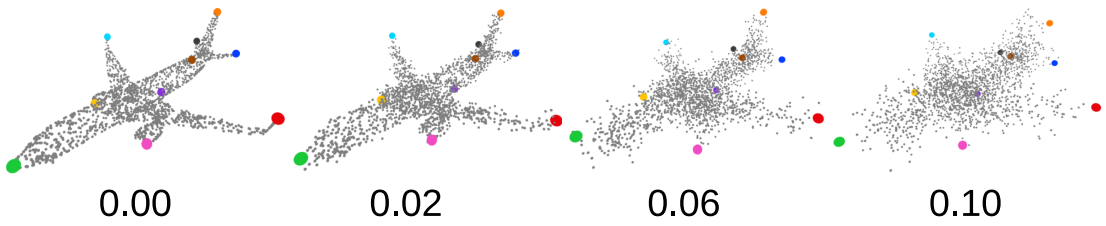
Table 6.4 Performance of the proposed approach for selected losses. Where ME represents matching error (coherence). The conditional formatting "green-to-red" represents the "good-to-bad" performance. The results are the average values of the test set of the keypointNet dataset.

w.o. loss	Inclusivity	Coverage	DAS	ME
All loss components	78.44	91.53	74.00	0.056
$\mathscr{L}_{overlap}$	77.09	90.72	53.80	0.061
$\mathscr{L}_{sep}$	63.01	85.70	67.38	0.081
$\mathscr{L}_{shape}$	76.05	90.31	58.45	0.064
$\mathscr{L}_{volume}$	77.35	90.90	63.14	0.066
$\mathscr{L}_{consist}$	76.52	91.03	42.44	0.103
$\mathscr{L}_{pose}$	76.76	91.04	53.89	0.095
$\mathscr{L}_{consist} + \mathscr{L}_{pose}$	70.15	88.07	41.95	0.103

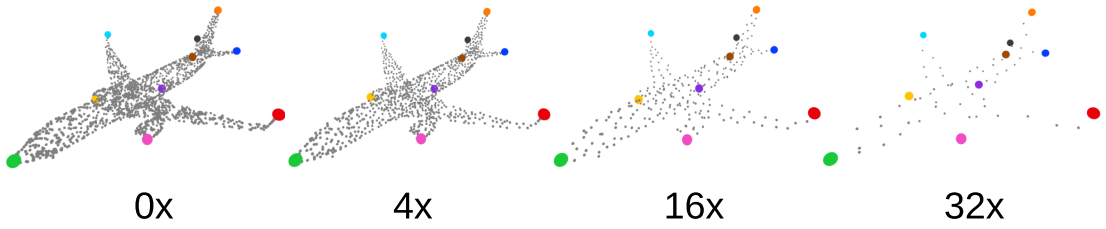
6.4.3 Robustness to perturbations

This ablation highlights the performance of the proposed network for noisy and down-sampled PCDs of the airplane category. Noisy PCDs are generated by adding Gaussian noise of different variances to the original PCDs. For decimating the PCD, we use the Farthest Point Sampling (FPS) as used in [69, 132] to sample points from original PCDs for different sampling ratios. Fig. 6.8a and Fig. 6.8b show the keypoints estimated for noisy and down-sampled PCDs, respectively. Here the network is able to successfully estimate the consistent keypoints at accurate positions for the noisy and down-sampled PCDs.

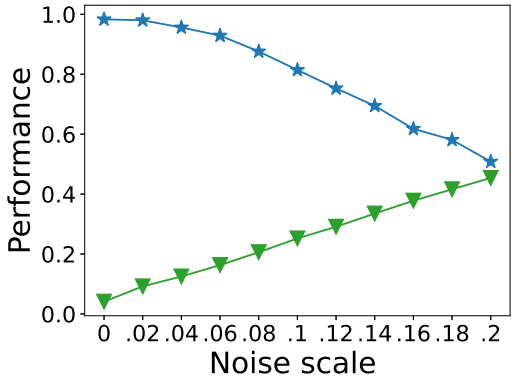
Quantitative results for the noisy and down-sampled PCDs are illustrated in Fig. 6.8c and Fig. 6.8d, respectively where to fix the DAS in the plots [0 to 1], we show DAS/100. The results show that the ME increases and the DAS decrease with the increase in the noise level (scale). Similarly, DAS decreases if down-sampling ratio is reduced to 6 times the original PCD. The ME remains approximately the same for down-sampled PCDs, validating that down-sampling does not affect the keypoints position. In the next section (Sec. 6.5), we also present quantitative results of the other categories for noisy and down-sampled PCDs. The presented visualizations show that the presented approach has successfully estimated keypoints for all the categories.



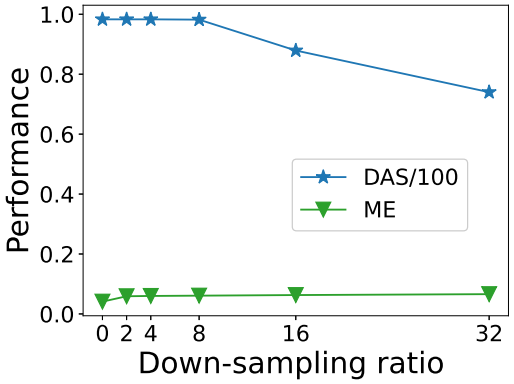
(a) Visualizations of the noisy PCDs



(b) Visualizations of the down-sampled PCDs



(c) Noise



(d) Down-sampling

Figure 6.8 Performance of the proposed approach for noisy and decimated PCDs. (a) and (b) represent qualitative results, whereas, (c) and (d) show plots illustrating the effect of the noisy and down-sampling PCDs, respectively.

6.5 Qualitative results

In this section, we present the keypoints estimated by the proposed SC3K for intra-class, noisy and down-sampled objects.

Qualitative Comparison with Intra-class objects:

The qualitative results of the experiments reported in Tab. 6.1 are shown in Fig. 6.9. The figure compares the keypoints estimated by the SC3K for intra-class objects. Four randomly selected objects are shown in the figure validating the fact that the keypoints are proximal to the original PCDs, semantically in order (coherent), and pointing to the sharp edges of the objects.

Visualisation of the noisy PCDs:

This section shows the qualitative results (extension of Fig. 6.8a) of the presented approach for different noisy PCDs. We add the Gaussian noise of different scales to the original PCDs of different categories. The noise scale is written in the beginning of every row where "0.00" mean original PCD without noise. The estimated keypoints are shown in Fig. 6.10. It can be observed that the proposed SC3K remains successful in estimating the 3D keypoints from the noisy PCDs. Moreover, the keypoints are always estimated close to the outermost points in the PCDs (i.e. close to the noisy surface). However, the accuracy decreases with the increase in the noise scale.

Visualisation of the Down-sampled PCDs:

This section presents the performance of our approach for down-sampled PCDs as an extension of the results shown in Fig. 6.8b. For decimating the PCD, we use the Farthest Point Sampling (FPS) as used in [132] to sample points from original PCDs for different sampling ratios. We test our pre-trained network to estimate the 3D keypoints from the down-sampled PCDs. The results are shown in Fig. 6.11. The figure is horizontally divided to fix all the objects on one page. Each column presents the results of a different object. The sampling ratio is shown at the beginning of every row. The "0×" shows the original PCD without sampling (zero times sampling). It can be observed that the SC3K has estimated approximately accurate keypoints for the down-sampled PCDs. However, the keypoints are not estimated at the same positions as the positions of the corresponding keypoints of the original PCDs (without sampling) when the PCDs are scaled 32 times (32×). The 32× sampling means a PCD containing only 64 points, considering that the original PCD contains 2048 points.

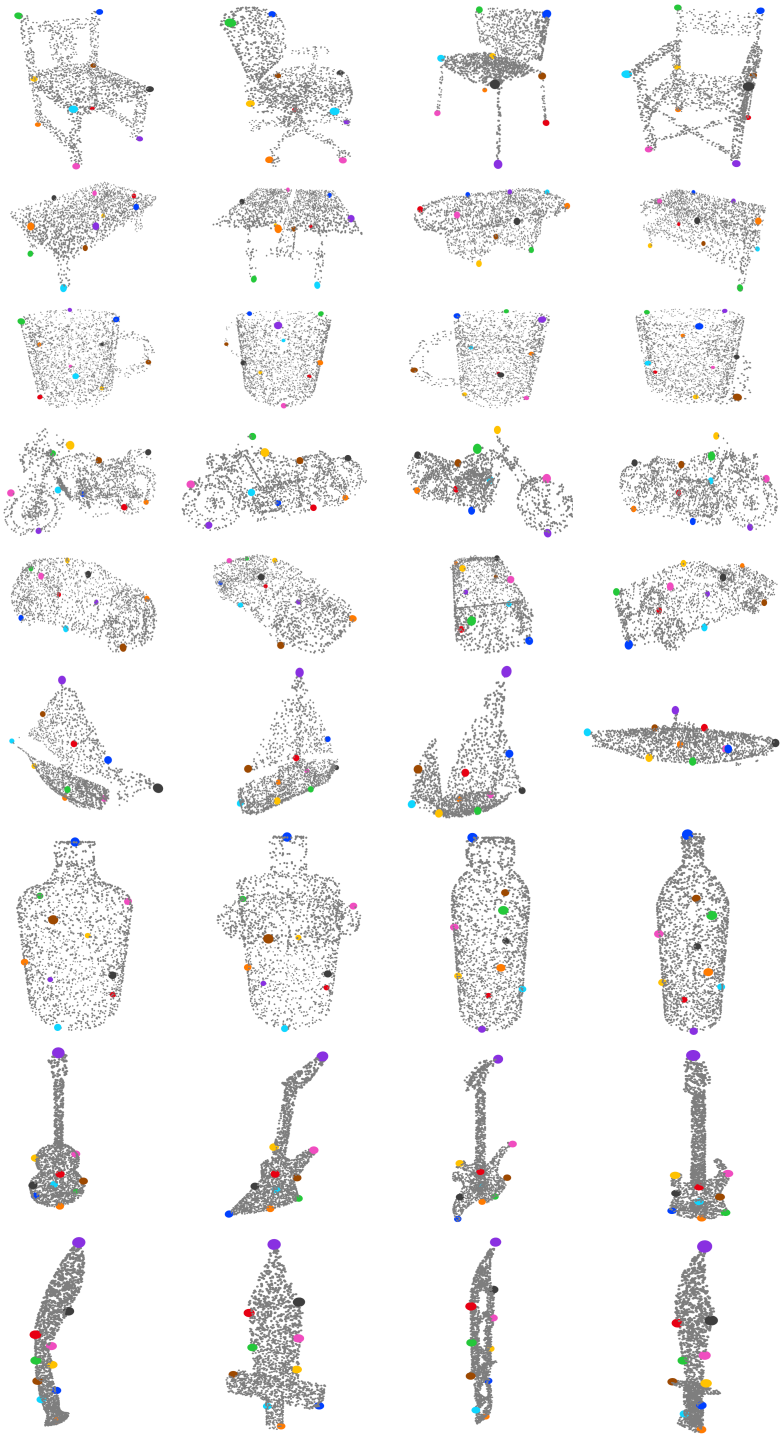


Figure 6.9 Qualitative results of the proposed SE3K for different categories. Every row shows four objects (in different poses) of the same category. The keypoints (coloured points) are estimated on the surface and in the same pose as the pose of the original PCDs (small gray points). Moreover, they are semantically consistent for all the intra-class objects.

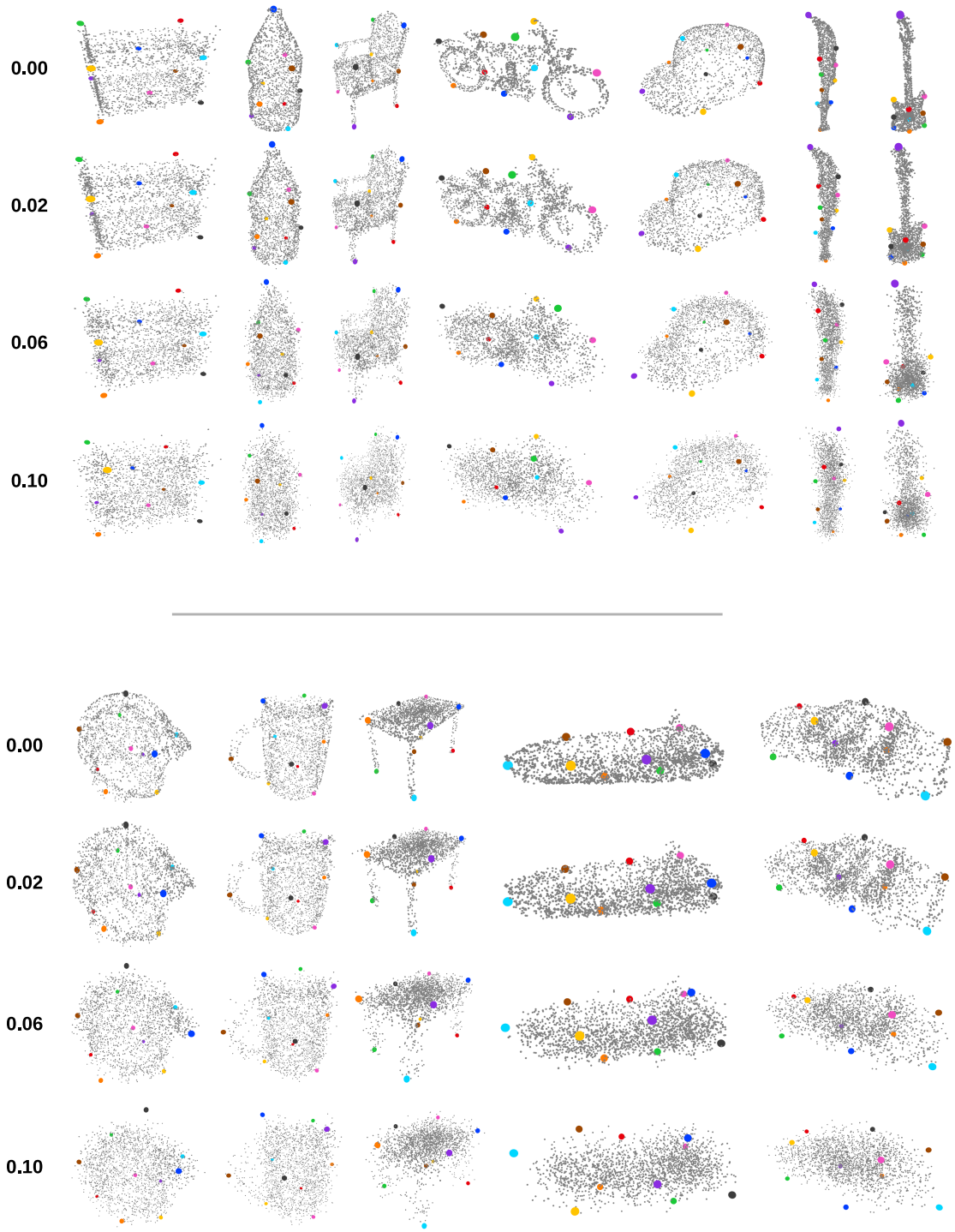


Figure 6.10 Performance of the proposed approach for the noisy PCDs. Gaussian noise of different scales (as mentioned at the beginning of every row) is added to the input PCDs. "0.00" represents the original PCD (without noise). The SC3K remains successful in estimating the semantically consistent keypoints for noisy PCDs. However, the accuracy has decreased with an increase in the noise scale.

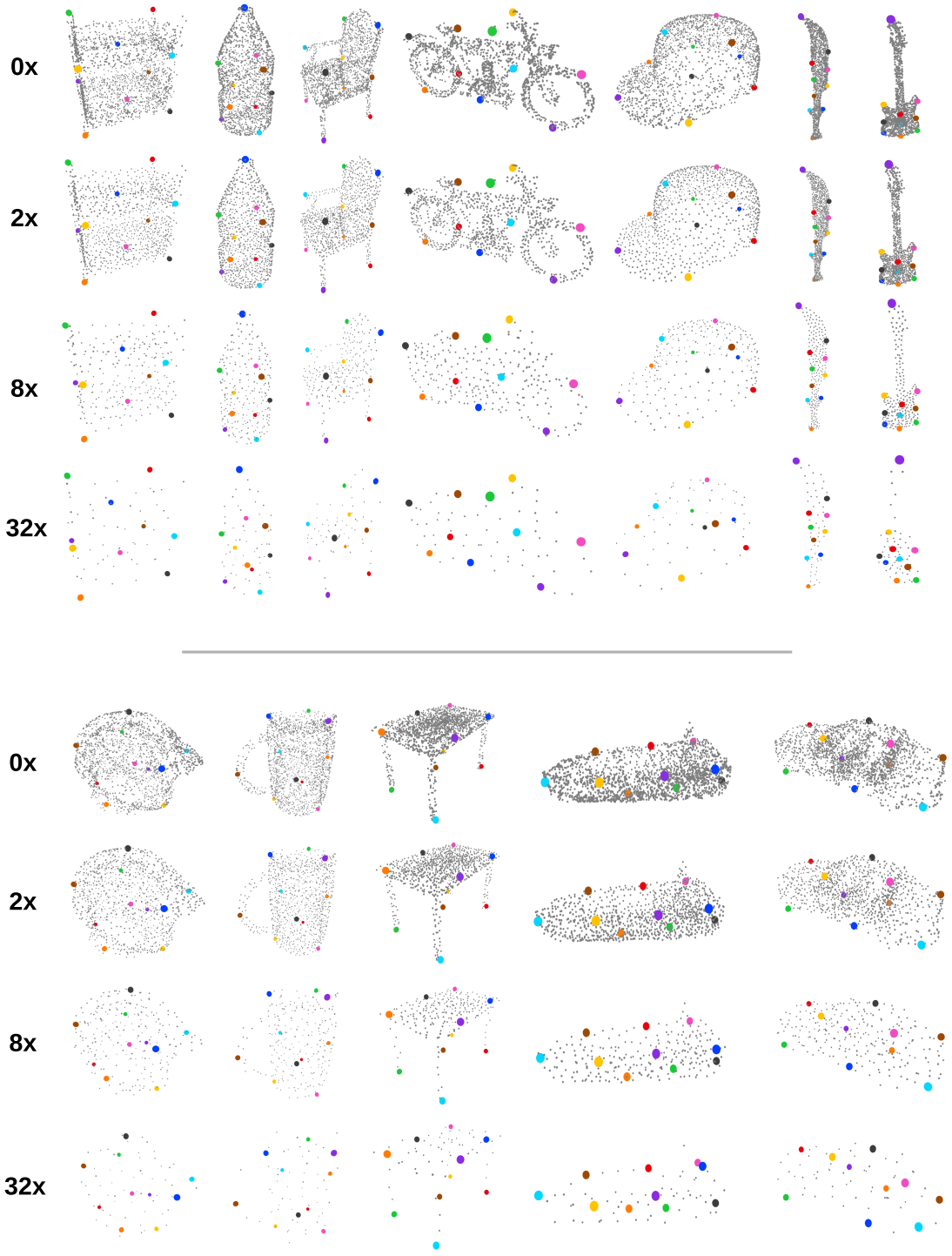


Figure 6.11 Performance of our method for down-sampled PCDs. The input PCDs are down-sampled for different scales, as mentioned at the beginning of every row. The "0×" shows the original PCDs. The proposed SC3K remains successful in estimating the approximately accurate 3D positions of the keypoints.

6.6 Chapter summary

The chapter presents a method to estimate 3D keypoints from a single PCD such that they express the following properties: *robust* – minimum position error across different rotated versions of the same PCD; *compact* – close or proximal to the PCD surface, *coherent* – in semantic order for all the intra-class instances. Similarly, the proposed method is *repeatable* – can estimate the accurate keypoints irrespective of the noise and down-sampling or rotation of the input PCD; and *self-supervised* – can estimate the same keypoints from single PCD without requiring any labels (pseudo or human annotation) during the inference. We achieve these desiderata by training the network with a new self-supervised strategy that does not require human annotations, instead, it computes the relative pose between the two sets of keypoints as a proxy task and then minimises the error against the known relative pose of the input PCDs pair. The proposed approach is compared with the SOTA keypoints estimation approaches using the KeypointNet dataset. The results validate that the presented approach outperforms the SOTA approaches by estimating the coherent keypoints close to the object's surface, characterising the object's shape.

There are two limitations of the presented approach: first, it may fail to estimate keypoints close to the object's surface for a number of keypoints higher than 35, and second, its performance may decrease for symmetrical shapes. For some categories, such as bikes or cars, it is challenging to differentiate between the front and back wheels. In the same way, the huge geometrical variation also negatively affects the performance, i.e. it's hard to compute semantically coherent keypoints between a single and a bunk bed. Our approach is dependent on several loss functions, so considering fewer loss components while achieving a similar performance can be considered a future task.

Chapter 7

Conclusions and Future Directions

In this chapter, we summarize the work presented in this thesis. Based on the nature of the addressed problems, we divide them into three sections: 3D shape reconstruction from a single-view RGB image, supervised keypoints estimation from a single-view RGB image and the self-supervised keypoints estimation from a PCD. In every section, we describe our solution(s) to the addressed problems, explain the limitations of the proposed solutions and suggest possible future directions. In the end, we summarize the thesis by presenting the conclusions.

7.1 3D shape reconstruction from a single-view RGB image

Proposed method: The first task presented in the thesis is related to 3D reconstruction. Considering the background information a basic limitation for the existing reconstruction approaches, in the task, we proposed a solution that estimated 3D keypoints from natural images in the presence of the real background. The proposed approach is compared with "Mesh R-CNN", which is considered as a State-Of-The-Art (SOTA) approach for 3D reconstruction from real images. Our approach outperforms the Mesh R-CNN by estimating the complete, smooth, and sharp 3D shape of an object.

Baseline solution: Moreover, we also present a baseline solution to execute existing approaches that are valid for only synthetic images – images with no/white background. The approach is based on two modules: segmenter and reconstructor. The segmenter module converts a real image into an image similar to the synthetic (single object in the center of a white background image) by exploiting a segmentation approach. The reconstructor module

uses the output of the segmenter (processed image) and reconstructs the 3D shape. Any approach that is valid only for synthetic images can be used as a reconstructor.

Limitations: Although the presented method allows estimation of more accurate 3D shape estimation, it has some limitations. First, it can reconstruct a single object at a time, i.e. the input image should contain only a single object. Second, the performance will be affected negatively if the object is moved away from the center of the image (translated). It is due to the fact that the network is trained for images those contain objects approximately around the center. However, rotation does not affect the performance since the network is trained for different randomly rotated objects. Third, the network does not estimate the object's pose, i.e., the reconstructed objects are always in the same canonical pose. Whereas for several downstream tasks, the pose information is required. Fifth, some of the parts of the objects are not accurately estimated due to different reasons. For example, sometimes it's difficult to separate an object from the surrounding (another object/background) due to the color combination or occlusions. Similarly, some of the parts are not so common across the objects in a category. Fig. 7.1 illustrates the execution of the SOTA approaches to highlight the inaccurate reconstruction of the leg joints and the back of the chair.

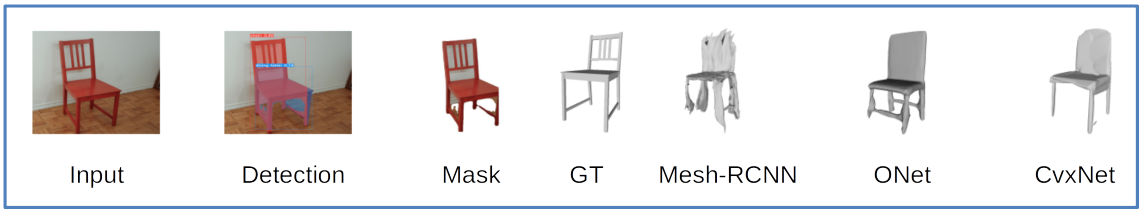


Figure 7.1 Execution of the Mesh R-CNN [31], ONet [68] and CvxNet [17]. The original input image is directly fed to the Mesh R-CNN. However, the masked version of the same image is used to test the ONet and CvxNet, because they work only for synthetic images with white backgrounds. It can be observed that the leg joints and the chair back are not reconstructed accurately by any of the approaches.

Future directions: Considering the above-mentioned limitations, the proposed approach can be improved in many ways. It can be extended for multiple objects' shape reconstruction from a single image by taking into account the working principle of Mesh R-CNN. For that, objects can be segmented from the image background using any segmentation algorithm. Then the reconstruction approach can be used to reconstruct all the separated objects. By doing this, the objects could be reconstructed in the canonical pose. However, to reconstruct the whole scene, the approach should also estimate the pose of every object in the image. This will allow the generation of the complete 3D scene by combining all the reconstructed

shapes with respect to their original pose. This future direction will also solve the translation limitation of the proposed approach. However, similar to the Mesh R-CNN, this solution will be based on the two modules, which is not an optimal end-to-end solution.

The pose of the reconstructed object can be estimated by adding another branch in the network that estimates the object's pose during training. However, to do so, we need the ground truth information of the pose of every object in the image. Another possible way is to estimate the 3D keypoint that can preserve the object's pose information and hence can be used to predict the relative pose between the estimated keypoints and the ground truth point clouds. The relative pose can be used to compute the transform of the estimated 3D shape during the training.

7.2 Supervised keypoints estimation from a single-view RGB image

Considering that the pose information in the 3D shape reconstruction task can be achieved using the keypoints, in this section, we aimed to estimate the keypoints from single-view RGB images in a supervised way. For this, we proposed two methods; first, without using a hallucination network and second, using the hallucination network for knowledge distillation. The details of both methods are as follows;

Proposed method 1: The approach takes an image as input and computes 3D keypoints in a fully supervised way. The locations of the estimated keypoints are corrected by comparing them with the corresponding ground truth keypoints. The loss, based on different components, is used to train the network. The estimated keypoints are used to compute the angular distance error between two views of an object. It is found error is comparatively lower than those of the SOTA approaches.

Proposed method 2: We observed that in method 1, we estimate the keypoints directly from the 2D features that are extracted from RGB images, so some of them (in some cases) are estimated outside the object. This can be improved by using 3D features that can be extracted from the original point clouds of the object to train the network. Therefore, as an extension, we present an upgraded version of method 1 that exploits the object's points clouds during training to learn to produce 3D features from RGB images that are simial to those extracted from the point clouds. During inference, the method uses only RGB images, by removing the need of point clouds, to extract 2D and 3D features that are later used to

estimate 3D keypoints. The proposed approaches are compared with the SOTA approaches that estimate 3D keypoints from images. It is found that the upgraded version (method 2) outperforms them by estimating keypoints that are comparatively good for computing relative pose between different views of an object.

Limitations of both methods: Although the network architectures and training procedures of both the presented methods are different, the overall main functionality is the same. Therefore, their limitations are the same as described below. First, they are fully dependent on the ground truth keypoints, thus, are applicable to a limited number of categories. It is due to the fact that annotating keypoints for every object is a hard task and requires great human assistance. That is why there are very limited human-annotated datasets are present. Second, in a similar way, the approaches estimate keypoints from synthetic images only and may fail to provide good keypoints for real images. It is due to the fact that they are trained for synthetic images containing objects of the same size positioned always in the center. Third, the approach considers only one object at a time that is approximately present close to the center of the object. This means that it may fail to produce good keypoints if the object is located far from the center. This limitation is also a barrier that does not allow the approaches to perform well on real images. Fourth, intra-class shape variations, such as single and bunk beds, make the problem more challenging. The network fails to estimate optimal keypoints in some similar scenarios. Fifth, the symmetry within the object's parts also reduces the performance of the proposed approaches. It is due to the fact that it is very hard to differentiate between the front and back wheels of the bike or car. The problem becomes more difficult when we consider object pose (rotated object), as we are considering, in our case, by rendering the objects in 24 different views. For two 180° rotated versions of the same object, the network can mix the front and the back wheel and hence can estimate semantically incorrect keypoints (i.e., the keypoints of the front wheel can be estimated on the back wheel).

Future directions: Considering the above-mentioned limitations, the presented approach can be extended in the following ways. Depending on a real dataset that contains pairs of an image and the point cloud of the object along with the corresponding 3D keypoints annotations, the approach can be trained to estimate 3D keypoints of an object from real images, i.e. images containing different sizes of objects at different positions on the image. Similarly, tackling multiple objects at the same time using some segmentation techniques can be considered as a future task. Moreover, the intra-class variations are difficult to consider by modifying the architecture. Adding more samples to the dataset or using augmentation

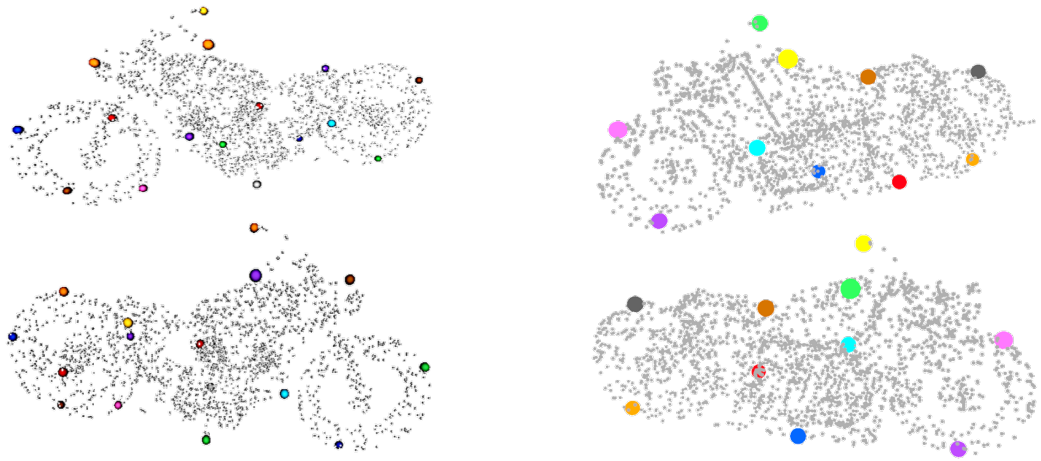
techniques to generate similar samples can be helpful in this case. It can also be solved by considering it as a domain shift problem. In the same way, considering symmetry is a challenging task. In real scenarios, the surrounding objects or background can play a helpful role in differentiating between the symmetric parts of any object.

7.3 Self-supervised keypoints estimation from a PCD

Proposed method: As a fourth task in the thesis, we present an approach that estimates 3D keypoints in a self-supervised way using the object's point clouds. Since the approach does not use any ground truth information for localizing the keypoints, their estimations are different than those estimated using the supervised approaches. The keypoints instead of highlighting the ground truth keypoints positions, highlight the shape of the original object. The approach is compared with SOTA approaches which use PCD to estimate the same keypoints. The results show that the presented approach outperforms by estimating keypoints that best characterize the shape of the object. Moreover, it is also shown that the approach can estimate accurate keypoints even for rotated, noisy and decimated PCD.

Limitations: The proposed approach has the following five limitations. First, the huge geometrical variation and the symmetrical parts of an object negatively affect the performance of the proposed approach. This limitation generally exists for all the (supervised and un-/self-supervised) keypoints estimation approaches. It is due to the fact that the object's same structured and symmetrically same parts (circular types of bikes or cars) are very difficult to identify. An example of a semantically wrong vs correct estimation of the keypoints for the symmetric parts of an object is shown in Fig. 7.2. Fig. 7.2a shows that the approach has localized the keypoints close to the object, however, their semantic order is wrong. The keypoints that are estimated on the front wheel for the original object (row 1) are estimated on the back wheel after the 180° rotated version (row 2). Whereas Fig. 7.2b shows that the keypoints are estimated with correct semantic information for both the versions, before and after the rotation. The difference in the semantic colors of the keypoints is due to the fact that the outputs are generated from two networks that are trained separately in different settings (e.g. number of keypoints).

Third, it may fail to estimate keypoints close to the object's surface for a number of keypoints higher than 35. Forth, the presented approach is dependent on several structural loss functions. Thus it is complex as it has to check many parameters during the training process. Fifth, the approach estimates fix a number of keypoints for all the objects. And it is trained separately



(a) Semantically wrong keypints (b) Semantically correct keypoints

Figure 7.2 Wrong (semantic) prediction for symmetric parts of an object when an object is rotated to for 180° due to symmetric parts of an object. (a) Wrong semantic information – the network could not differentiate the front and the back tyres and mixed the semantic order of the estimated keypoints. (b) Correct semantic information – the network remains successful in predicting the semantic order of the estimated keypoints for the symmetric parts of the object (tyres) even after 180° rotation.

for different categories considering that a fixed number of keypoints can not be used to represent objects of the different structures. This shows that there is a need to select a different number of keypoints with respect to the category.

Future directions: The first two limitations of this approach are exactly the same as those of other keypoints estimation approaches. Their possible solutions are described earlier. However, improving the network architecture such that it can (1) estimate keypoints higher than 35, (2) be trained for a different number of keypoints, and (3) be trained together for all the categories can be considered as a future task. The confidence scores, as estimated in the proposed supervised technique, can be used to solve the fourth and fifth limitations. However, for estimating the confidence scores, we need ground truth information which is not available in the self-supervised settings. The most significant improvement in the proposed approach would be the use of the minimum number of loss functions while achieving the same performance.

Furthermore, apart from the improvement in the proposed approach, there could be some other future directions. Such as keypoints estimation from the partial or noisy PCD (those similar to the LiDAR data). This allows the direct applicability of the approach to real scenarios. In a similar way, the keypoints can be used to track moving objects like an outdoor

car. Another direction could be the object shape completion using the object's symmetric parts and the corresponding keypoints.

7.4 Conclusions

In this thesis, we developed deep learning models to investigate problems associated with the 3D properties of an object. We start with the 3D shape reconstruction problem. We observed that some of the existing 3D reconstruction approaches are valid only for synthetic images, whereas others methods fail to produce complete, smooth and sharp 3D shapes from real objects. Considering the limitations, we proposed two solutions. The first solution enables the existing synthetic approaches to produce comparatively good 3D shapes from real images by segmenting an object from a background using a segmentation module. Although this solution improves the reconstruction accuracy, it can not be considered as an optimal solution. It is because the solution is not end-to-end, and the performance is dependent on the segmentation module. To overcome this limitation, we present an end-to-end solution that removes the need of the segmentation module and reconstructs the shape from real images by separating the features of the object from the features of the background. To separate the object's features, the network is trained using pairs of color and white background images. The proposed approach is compared with the SOTA approaches, which shows that it outperforms them by estimating smooth, sharp and more accurate 3D shapes.

We observed that the proposed solution for the 3D reconstruction problem always estimates the 3D shape in a canonical pose. The literature reports that keypoints can be used to estimate an object's pose. Thus, we considered keypoints estimation as the next task. We estimate the 3D keypoints in a fully supervised way by using ground truth keypoints that are available in the dataset. The presented approach uses only a single image to estimate the keypoints. The keypoints are used to estimate the relative pose between different views of an object, and it is found that it outperforms the existing SOTA approaches. However, since only the images are used to train the network, it estimates some of the keypoints outside the object in some cases.

we extended the previously proposed keypoints estimation approach and presented a teach-student network to distil knowledge from the point clouds during training. The network is trained in two steps; first, the teacher module is trained to extract 3D features from the point clouds, and then the student module learns from the teacher module to produce similar 3D features from images. During inference, only the images are used to extract 2D and 3D features that are later used to estimate 3D keypoints. The approach is compared with

the previously proposed approach and the other SOTA approaches that estimate keypoints from images. The proposed approach remains successful in estimating the keypoints that are comparatively more accurate for relative pose estimation.

The above proposed two approaches are fully supervised, and hence they require a huge dataset with human annotations. Creating such datasets is a hard task as it consumes a lot of time and requires a human assistant to label every object in a category. This is why limited datasets have been created so far. To overcome this problem, we present an approach that estimates 3D keypoints from PCDs in a self-supervised setting. The approach, by removing the need of the ground truth keypoints, increases its viability for a large number of datasets (such as the real datasets that contain decimated and noisy objects). The approach is compared with the existing un-/self-supervised approaches. It is found that our approach outperforms them by estimating keypoints that best characterize an object's shape, even for rotated, decimated and noisy PCDs.

References

[1] Abate, A. F., Bisogni, C., Castiglione, A., and Nappi, M. (2022). Head pose estimation: An extensive survey on recent techniques and applications. *Pattern Recognition*, 127:108591.

[2] Adamczyk, D. and Hula, J. (2020). Keypoints selection using evolutionary algorithms. In *ITAT*, pages 186–191.

[3] Barabanau, I., Artemov, A., Burnaev, E., and Murashkin, V. (2020). Monocular 3d object detection via geometric reasoning on keypoints. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 652–659.

[4] Bin, Y., Chen, Z.-M., Wei, X.-S., Chen, X., Gao, C., and Sang, N. (2020). Structure-aware human pose estimation with graph convolutional networks. *Pattern Recognition*, 106:107410.

[5] Bisio, I., Haleem, H., Garibotto, C., Lavagetto, F., and Sciarrone, A. (2021). Performance evaluation and analysis of drone-based vehicle detection techniques from deep learning perspective. *IEEE Internet of Things Journal*.

[6] Bojanić, D., Bartol, K., Petković, T., and Pribanić, T. (2020). A review of rigid 3d registration methods. In *13th International Scientific-Professional Symposium Textile Science and Economy*, pages 286–296.

[7] Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J. (2019). Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9157–9166.

[8] Cao, H., Xu, Y., Yang, J., Mao, K., Yin, J., and See, S. (2021). Effective action recognition with embedded key point shifts. *Pattern Recognition*, 120:108172.

[9] Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. (2015). Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

[10] Chen, H., Sun, D., Liu, W., Wu, H., Liang, M., and Liu, P. X. (2022). A novel approach to the extraction of key points from 3d rigid point cloud using 2d images transformation. *IEEE Transactions on Geoscience and Remote Sensing*.

[11] Chen, N., Liu, L., Cui, Z., Chen, R., Ceylan, D., Tu, C., and Wang, W. (2020a). Unsupervised learning of intrinsic structural representation points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9121–9130.

[12] Chen, Z., Tagliasacchi, A., and Zhang, H. (2020b). Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 45–54.

[13] Chen, Z. and Zhang, H. (2019). Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948.

[14] Cheng, A.-C., Li, X., Sun, M., Yang, M.-H., and Liu, S. (2021a). Learning 3d dense correspondence via canonical point autoencoder. *Advances in Neural Information Processing Systems*, 34:6608–6620.

[15] Cheng, S., Chen, X., He, X., Liu, Z., and Bai, X. (2021b). Pra-net: Point relation-aware network for 3d point cloud analysis. *IEEE Transactions on Image Processing*, 30:4436–4448.

[16] Choy, C. B., Xu, D., Gwak, J., Chen, K., and Savarese, S. (2016). 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer.

[17] Deng, B., Genova, K., Yazdani, S., Bouaziz, S., Hinton, G., and Tagliasacchi, A. (2020). Cvxnet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 31–44.

[18] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[19] Devgon, S., Ichnowski, J., Balakrishna, A., Zhang, H., and Goldberg, K. (2020). Orienting novel 3d objects using self-supervised learning of rotation transforms. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1453–1460. IEEE.

[20] Duan, Y., He, K., Feng, J., Lu, J., and Zhou, J. (2022). Estimating 3d finger pose via 2d-3d fingerprint matching. In *27th International Conference on Intelligent User Interfaces*, pages 459–469.

[21] Everingham, M., Eslami, S., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136.

[22] Fan, Z., Hu, X., Chen, C., Wang, X., and Peng, S. (2023). A landmark-free approach for automatic, dense and robust correspondence of 3d faces. *Pattern Recognition*, 133:108971.

[23] Feng, Q., Luo, Y., Luo, K., and Yang, Y. (2021). Look, cast and mold: learning 3d shape manifold from single-view synthetic data. *arXiv preprint arXiv:2103.04789*.

[24] Fernandez-Labrador, C., Chhatkuli, A., Paudel, D. P., Guerrero, J. J., Demonceaux, C., and Gool, L. V. (2020). Unsupervised learning of category-specific symmetric 3d keypoints from point sets. In *European Conference on Computer Vision*, pages 546–563. Springer.

[25] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

[26] Garcia, N. C., Morerio, P., and Murino, V. (2018). Modality distillation with multiple stream networks for action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118.

[27] Garcia, N. C., Morerio, P., and Murino, V. (2019). Learning with privileged information via adversarial discriminative modality distillation. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2581–2593.

[28] Georgakis, G., Karanam, S., Wu, Z., and Kosecka, J. (2019). Learning local rgb-to-cad correspondences for object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8967–8976.

[29] Girdhar, R., Fouhey, D. F., Rodriguez, M., and Gupta, A. (2016). Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer.

[30] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

[31] Gkioxari, G., Malik, J., and Johnson, J. (2019). Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9785–9795.

[32] Gu, J., Ma, W.-C., Manivasagam, S., Zeng, W., Wang, Z., Xiong, Y., Su, H., and Urtasun, R. (2020). Weakly-supervised 3d shape completion in the wild. In *European Conference on Computer Vision*, pages 283–299. Springer.

[33] Gupta, K., Jabbireddy, S., Shah, K., Shrivastava, A., and Zwicker, M. (2020). Improved modeling of 3d shapes with multi-view depth maps. In *2020 International Conference on 3D Vision (3DV)*, pages 71–80. IEEE.

[34] Han, K., Rezende, R. S., Ham, B., Wong, K.-Y. K., Cho, M., Schmid, C., and Ponce, J. (2017). Scnet: Learning semantic correspondence. In *Proceedings of the IEEE international conference on computer vision*, pages 1831–1840.

[35] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

[36] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[37] He, Y., Sun, W., Huang, H., Liu, J., Fan, H., and Sun, J. (2020). Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11632–11641.

[38] Hoffman, J., Gupta, S., and Darrell, T. (2016). Learning with side information through modality hallucination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 826–834.

[39] Hung, C.-W., Chang, R.-C., Chen, H.-S., Liang, C. H., Chan, L., and Chen, B.-Y. (2022). Puppeteer: Exploring intuitive hand gestures and upper-body postures for manipulating human avatar actions. In *28th ACM Symposium on Virtual Reality Software and Technology*, pages 1–11.

[40] Iqbal, U., Xie, K., Guo, Y., Kautz, J., and Molchanov, P. (2021). Kama: 3d keypoint aware body mesh articulation. In *2021 International Conference on 3D Vision (3DV)*, pages 689–699. IEEE.

[41] Jakab, T., Tucker, R., Makadia, A., Wu, J., Snavely, N., and Kanazawa, A. (2021). Keypointdeformer: Unsupervised 3d keypoint discovery for shape control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12783–12792.

[42] Kanazawa, A., Black, M. J., Jacobs, D. W., and Malik, J. (2018). End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7122–7131.

[43] Kao, Y., Li, W., Wang, Q., Lin, Z., Kim, W., and Hong, S. (2020). Synthetic depth transfer for monocular 3d object pose estimation in the wild. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11221–11228.

[44] Kaya, B. and Timofte, R. (2020). Self-supervised 2d image to 3d shape translation with disentangled representations. In *2020 International Conference on 3D Vision (3DV)*, pages 1039–1048. IEEE.

[45] Khan, M. S. U., Pagani, A., Liwicki, M., Stricker, D., and Afzal, M. Z. (2022). Three-dimensional reconstruction from a single rgb image using deep learning: A review. *Journal of Imaging*, 8(9):225.

[46] Kong, C., Lin, C.-H., and Lucey, S. (2017). Using locally corresponding cad models for dense 3d reconstructions from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4857–4865.

[47] Li, C., Zeeshan Zia, M., Tran, Q.-H., Yu, X., Hager, G. D., and Chandraker, M. (2017). Deep supervision with shape concepts for occlusion-aware 3d object parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5465–5474.

[48] Li, J. and Lee, G. H. (2019). Usip: Unsupervised stable interest point detection from 3d point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 361–370.

[49] Li, P., Chen, X., and Shen, S. (2019). Stereo r-cnn based 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7644–7652.

[50] Li, Y., Torralba, A., Anandkumar, A., Fox, D., and Garg, A. (2020). Causal discovery in physical systems from videos. *Advances in Neural Information Processing Systems*, 33:9180–9192.

[51] Li, Z., Yao, Y., Quan, Z., Xie, J., and Yang, W. (2022). Spatial information enhancement network for 3d object detection from point cloud. *Pattern Recognition*, 128:108684.

[52] Liao, Z., Yang, J., Saito, J., Pons-Moll, G., and Zhou, Y. (2022). Skeleton-free pose transfer for stylized 3d characters. In *European Conference on Computer Vision*, pages 640–656. Springer.

[53] Lin, S., Wang, Z., Ling, Y., Tao, Y., and Yang, C. (2022a). E2ek: End-to-end regression network based on keypoint for 6d pose estimation. *IEEE Robotics and Automation Letters*, 7(3):6526–6533.

[54] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

[55] Lin, Y., Chen, L., Huang, H., Ma, C., Han, X., and Cui, S. (2021). Beyond farthest point sampling in point-wise analysis. *arXiv preprint arXiv:2107.04291*.

[56] Lin, Y., Chen, L., Huang, H., Ma, C., Han, X., and Cui, S. (2022b). Task-aware sampling layer for point-wise analysis. *IEEE Transactions on Visualization and Computer Graphics*.

[57] Liu, H., Cong, Y., Yang, C., and Tang, Y. (2019). Efficient 3d object recognition via geometric information preservation. *Pattern Recognition*, 92:135–145.

[58] Liu, L., Yang, L., Chen, W., and Gao, X. (2021). Dual-view 3d human pose estimation without camera parameters for action recognition. *IET Image Processing*, 15(14):3433–3440.

[59] Liu, Q., Zhang, Y., Bai, S., and Yuille, A. (2022). Explicit occlusion reasoning for multi-person 3d human pose estimation. In *European Conference on Computer Vision*, pages 497–517. Springer.

[60] Liu, X., Jonschkowski, R., Angelova, A., and Konolige, K. (2020a). Keypose: Multi-view 3d labeling and keypoint estimation for transparent objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11602–11610.

[61] Liu, Z., Wu, Z., and Tóth, R. (2020b). Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 996–997.

[62] Lombardi, S., Oswald, M. R., and Pollefeys, M. (2020). Scalable point cloud-based reconstruction with local implicit functions. In *2020 International Conference on 3D Vision (3DV)*, pages 997–1007. IEEE.

[63] Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169.

[64] Lu, J., Richter, F., and Yip, M. (2020). Robust keypoint detection and pose estimation of robot manipulators with self-occlusions via sim-to-real transfer. *arXiv preprint arXiv:2010.08054*.

[65] Lu, J., Richter, F., and Yip, M. C. (2022). Pose estimation for robot manipulators via keypoint optimization and sim-to-real transfer. *IEEE Robotics and Automation Letters*, 7(2):4622–4629.

[66] Mariotti, O., Mac Aodha, O., and Bilen, H. (2021). Viewnet: Unsupervised viewpoint estimation from conditional generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10418–10428.

[67] Mei, G., Yu, L., Wu, Q., Zhang, J., and Bennamoun, M. (2022). Unsupervised learning on 3d point clouds by clustering and contrasting. *arXiv preprint arXiv:2202.02543*.

[68] Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., and Geiger, A. (2019). Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470.

[69] Mohammadi, S. S., Wang, Y., and Del Bue, A. (2021). Pointview-gcn: 3d shape classification with multi-view point clouds. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3103–3107. IEEE.

[70] Nenglun, C., Lingjie, L., Zhiming, C., et al. (2020). Unsupervised learning of intrinsic structural representation points. In *Conference on Computer Vision and Pattern Recognition*, pages 9118–9127.

[71] Oh Song, H., Xiang, Y., Jegelka, S., and Savarese, S. (2016). Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012.

[72] Paoletti, G., Cavazza, J., Beyan, C., and Del Bue, A. (2021). Subspace clustering for action recognition with covariance representations and temporal pruning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6035–6042. IEEE.

[73] Paoletti, G., Cavazza, J., Beyan, C., and Del Bue, A. (2022). Unsupervised human action recognition with skeletal graph laplacian and self-supervised viewpoints invariance. *arXiv preprint arXiv:2204.10312*.

[74] Park, S., Lee, M., and Kwak, N. (2020). Procrustean regression networks: Learning 3d structure of non-rigid objects from 2d annotations. In *European Conference on Computer Vision*, pages 1–18. Springer.

[75] Pinheiro, P. O., Rostamzadeh, N., and Ahn, S. (2019). Domain-adaptive single-view 3d reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7638–7647.

[76] Pons, J.-P., Keriven, R., and Faugeras, O. (2005). Modelling dynamic scenes by registering multi-view image sequences. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 822–827. IEEE.

[77] Poursaeed, O., Jiang, T., Qiao, H., Xu, N., and Kim, V. G. (2020). Self-supervised learning of point clouds via orientation estimation. In *2020 International Conference on 3D Vision (3DV)*, pages 1018–1028. IEEE.

[78] Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660.

[79] Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

[80] Robson, M. and Sridharan, M. (2022). Generating task-specific robotic grasps. *arXiv preprint arXiv:2203.10498*.

[81] Rochow, A., Schwarz, M., Schreiber, M., and Behnke, S. (2022). Vr facial animation for immersive telepresence avatars. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*.

[82] Ross, A. (2004). Procrustes analysis. *Course report, Department of Computer Science and Engineering, University of South Carolina*, 26:1–8.

[83] Sahin, C. (2022). Cmd-net: Self-supervised category-level 3d shape denoising through canonicalization. *Applied Sciences*, 12(20):10474.

[84] Sajnani, R., Poulenard, A., Jain, J., Dua, R., Guibas, L. J., and Sridhar, S. (2022). Condor: Self-supervised canonicalization of 3d pose for partial shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16969–16979.

[85] Salvi, A., Gavenski, N., Pooch, E., Tasoniero, F., and Barros, R. (2020). Attention-based 3d object reconstruction from a single image. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[86] Schwarz, M., Lenz, C., Rochow, A., Schreiber, M., and Behnke, S. (2021). Nimbro avatar: Interactive immersive telepresence with force-feedback telemanipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5312–5319. IEEE.

[87] Shah, S. A. A., Bennamoun, M., and Boussaid, F. (2017). Keypoints-based surface representation for 3d modeling and 3d object recognition. *Pattern Recognition*, 64:29–38.

[88] Shen, X., Wang, C., Li, X., Peng, Y., He, Z., Wen, C., and Cheng, M. (2022). Learning scale awareness in keypoint extraction and description. *Pattern Recognition*, 121:108221.

[89] Shi, R., Xue, Z., You, Y., and Lu, C. (2021a). Skeleton merger: an unsupervised aligned keypoint detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 43–52.

[90] Shi, Y., Ni, B., Liu, J., Rong, D., Qian, Y., and Zhang, W. (2021b). Geometric granularity aware pixel-to-mesh. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13097–13106.

[91] Shu, Z., Yang, S., Xin, S., Pang, C., Jin, X., Kavan, L., and Liu, L. (2021). Detecting 3d points of interest using projective neural networks. *IEEE Transactions on Multimedia*, 24:1637–1650.

[92] Sipiran, I. and Bustos, B. (2011). Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27(11):963–976.

[93] Spezialetti, R., Salti, S., and Di Stefano, L. (2021). Performance evaluation of 3d descriptors paired with learned keypoint detectors. *AI*, 2(2):229–243.

[94] Spezialetti, R., Tan, D. J., Tonioni, A., Tateno, K., and Tombari, F. (2020). A divide et impera approach for 3d shape reconstruction from multiple views. In *2020 International Conference on 3D Vision (3DV)*, pages 160–170. IEEE.

[95] Sun, W., Tagliasacchi, A., Deng, B., Sabour, S., Yazdani, S., Hinton, G. E., and Yi, K. M. (2021). Canonical capsules: Self-supervised capsules in canonical pose. *Advances in Neural Information Processing Systems*, 34:24993–25005.

[96] Sun, X., Huang, Y., and Lian, Z. (2022). Learning isometry-invariant representations for point cloud analysis. *Pattern Recognition*, page 109087.

[97] Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J. B., and Freeman, W. T. (2018). Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2974–2983.

[98] Suwajanakorn, S., Snavely, N., Tompson, J. J., and Norouzi, M. (2018). Discovery of latent 3d keypoints via end-to-end geometric reasoning. *Advances in neural information processing systems*, 31.

[99] Tang, J., Gong, Z., Yi, R., Xie, Y., and Ma, L. (2022a). Lake-net: Topology-aware point cloud completion by localizing aligned keypoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1726–1735.

[100] Tang, J., Xu, J., Gong, J., Song, H., Xie, Y., and Ma, L. (2022b). Prototype-aware heterogeneous task for point cloud completion. *arXiv preprint arXiv:2209.01733*.

[101] Tang, R., Wang, L., and Guo, Z. (2021). A multi-task neural network for action recognition with 3d key-points. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3899–3906. IEEE.

[102] Thakur, S. K., Beyan, C., Morerio, P., and Del Bue, A. (2021). Predicting gaze from egocentric social interaction videos and imu data. In *Proceedings of the 2021 International Conference on Multimodal Interaction*, pages 717–722.

[103] Uppal, H., Sepas-Moghaddam, A., Greenspan, M., and Etemad, A. (2021). Teacher-student adversarial depth hallucination to improve face recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3671–3680.

[104] Vecerik, M., Regli, J.-B., Sushkov, O., Barker, D., Pevceviciute, R., Rothörl, T., Schuster, C., Hadsell, R., Agapito, L., and Scholz, J. (2021). S3k: Self-supervised semantic keypoints for robotic manipulation via multi-view consistency. In *Proceedings of the 2020 Conference on Robot Learning*, pages 449–460.

[105] Wan, C., Probst, T., Gool, L. V., and Yao, A. (2019). Self-supervised 3d hand pose estimation through training by fitting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10853–10862.

[106] Wandt, B., Rudolph, M., Zell, P., Rhodin, H., and Rosenhahn, B. (2021). Canonpose: Self-supervised monocular 3d human pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13294–13304.

[107] Wang, C., Martín-Martín, R., Xu, D., Lv, J., Lu, C., Fei-Fei, L., Savarese, S., and Zhu, Y. (2020a). 6-pack: Category-level 6d pose tracker with anchor-based keypoints. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10059–10066. IEEE.

[108] Wang, H., Guo, J., Yan, D.-M., Quan, W., and Zhang, X. (2018a). Learning 3d keypoint descriptors for non-rigid shape matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19.

[109] Wang, M., Sun, C., and Sowmya, A. (2022). Complex shearlets and rotary phase congruence tensor for corner detection. *Pattern Recognition*, 128:108606.

[110] Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y.-G. (2018b). Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67.

[111] Wang, Y. and Solomon, J. M. (2019). Prnet: Self-supervised learning for partial-to-partial registration. *Advances in neural information processing systems*, 32.

[112] Wang, Z., Isler, V., and Lee, D. D. (2020b). Surface hof: surface reconstruction from a single image using higher order function networks. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 2666–2670. IEEE.

[113] Wei, G., Ma, L., Wang, C., Desrosiers, C., and Zhou, Y. (2021). Multi-task joint learning of 3d keypoint saliency and correspondence estimation. *Computer-Aided Design*, 141:103105.

[114] Wu, J., Wang, Y., Xue, T., Sun, X., Freeman, B., and Tenenbaum, J. (2017). Marrnet: 3d shape reconstruction via 2.5 d sketches. *Advances in neural information processing systems*, 30.

[115] Wu, J., Zhang, C., Zhang, X., Zhang, Z., Freeman, W. T., and Tenenbaum, J. B. (2018). Learning shape priors for single-view 3d completion and reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 646–662.

[116] Wu, R., Zhuang, Y., Xu, K., Zhang, H., and Chen, B. (2020a). Pq-net: A generative part seq2seq network for 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 829–838.

[117] Wu, W., Zhang, Y., Wang, D., and Lei, Y. (2020b). Sk-net: Deep learning on point cloud via end-to-end discovery of spatial keypoints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6422–6429.

[118] Xiang, Y., Mottaghi, R., and Savarese, S. (2014). Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, pages 75–82. IEEE.

[119] Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE.

[120] Xie, H., Yao, H., Zhang, S., Zhou, S., and Sun, W. (2020). Pix2vox++: Multi-scale context-aware 3d object reconstruction from single and multiple images. *International Journal of Computer Vision*, 128(12):2919–2935.

[121] Xue, H., Liu, L., Xu, W., Fu, H., and Lu, C. (2021). Omad: Object model with articulated deformations for pose estimation and retrieval. *arXiv preprint arXiv:2112.07334*.

[122] Yamashita, K., Nobuhara, S., and Nishino, K. (2019). 3dgmnet: Learning to estimate 3d shape from a single image as a gaussian mixture. *arXiv preprint arXiv:1912.04663*.

[123] Yang, B., Rosa, S., Markham, A., Trigoni, N., and Wen, H. (2018). Dense 3d object reconstruction from a single depth view. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):2820–2834.

[124] Yang, S., Xu, M., Xie, H., Perry, S., and Xia, J. (2021). Single-view 3d object reconstruction from shape priors in memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3152–3161.

[125] Yi, L., Su, H., Guo, X., and Guibas, L. J. (2017). Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2282–2290.

[126] Yin, K., Chen, Z., Chaudhuri, S., Fisher, M., Kim, V. G., and Zhang, H. (2020). Coalesce: Component assembly by learning to synthesize connections. In *2020 International Conference on 3D Vision (3DV)*, pages 61–70. IEEE.

[127] You, Y., Li, C., Lou, Y., Cheng, Z., Li, L., Ma, L., Wang, W., and Lu, C. (2019). Fine-grained object semantic understanding from correspondences. *arXiv preprint arXiv:1912.12577*.

[128] You, Y., Liu, W., Li, Y.-L., Wang, W., and Lu, C. (2020a). Ukpgan: Unsupervised keypoint ganeration. *arXiv preprint arXiv:2011.11974*.

[129] You, Y., Liu, W., Ze, Y., Li, Y.-L., Wang, W., and Lu, C. (2022). Ukpgan: A general self-supervised keypoint detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17042–17051.

[130] You, Y., Lou, Y., Li, C., Cheng, Z., Li, L., Ma, L., Lu, C., and Wang, W. (2020b). Keypointnet: A large-scale 3d keypoint dataset aggregated from numerous human annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13647–13656.

[131] Yuan, H., Zhao, C., Fan, S., Jiang, J., and Yang, J. (2022). Unsupervised learning of 3d semantic keypoints with mutual reconstruction. *arXiv preprint arXiv:2203.10212*.

[132] Yuan, W., Khot, T., Held, D., Mertz, C., and Hebert, M. (2018). Pcn: Point completion network. In *International Conference on 3D Vision*, pages 728–737. IEEE.

[133] Yuan, Y., Borrmann, D., Hou, J., Ma, Y., Nüchter, A., and Schwertfeger, S. (2021a). Self-supervised point set local descriptors for point cloud registration. *Sensors*, 21(2):486.

[134] Yuan, Y., Wei, S.-E., Simon, T., Kitani, K., and Saragih, J. (2021b). Simpoe: Simulated character control for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7159–7169.

[135] Zhang, C., Zhan, F., and Chang, Y. (2021). Deep monocular 3d human pose estimation via cascaded dimension-lifting. *arXiv preprint arXiv:2104.03520*.

[136] Zhang, P., Xie, R., Sun, J., Li, W., and Su, Z. (2022a). Au-pd: An arbitrary-size and uniform downsampling framework for point clouds. *arXiv preprint arXiv:2211.01110*.

[137] Zhang, Y., Huo, K., Liu, Z., Zang, Y., Liu, Y., Li, X., Zhang, Q., and Wang, C. (2020). Pgnet: A part-based generative network for 3d object reconstruction. *Knowledge-Based Systems*, 194:105574.

[138] Zhang, Z., Sun, J., Dai, Y., Zhou, D., Song, X., and He, M. (2022b). Self-supervised rigid transformation equivariance for accurate 3d point cloud registration. *Pattern Recognition*, 130:108784.

[139] Zhao, H., Tang, M., and Ding, H. (2020a). Hoppf: A novel local surface descriptor for 3d object recognition. *Pattern Recognition*, 103:107272.

[140] Zhao, W., Zhang, S., Guan, Z., Zhao, W., Peng, J., and Fan, J. (2020b). Learning deep network for detecting 3d object keypoints and 6d poses. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 14134–14142.

[141] Zhao, Y., Fang, G., Guo, Y., Guibas, L., Tombari, F., and Birdal, T. (2022). 3dpointcaps++: Learning 3d representations with capsule networks. *International Journal of Computer Vision*, 130(9):2321–2336.

[142] Zheng, Z., Yu, T., Dai, Q., and Liu, Y. (2021). Deep implicit templates for 3d shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1429–1439.

[143] Zhong, C., You, P., Chen, X., Zhao, H., Sun, F., Zhou, G., Mu, X., Gan, C., and Huang, W. (2022). Snake: Shape-aware neural 3d keypoint field. *arXiv preprint arXiv:2206.01724*.

[144] Zhong, Y. (2009). Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *2009 IEEE 12th international conference on computer vision workshops, ICCV Workshops*, pages 689–696. IEEE.

[145] Zhou, X. and et. al. (2018). Starmap for category-agnostic keypoint and viewpoint estimation. *ECCV*, pages 318–334.

[146] Zhou, Y., Liu, S., and Ma, Y. (2020). Learning to detect 3d reflection symmetry for single-view reconstruction. *arXiv preprint arXiv:2006.10042*.

[147] Zohaib, M., Taiana, M., and Bue, A. D. (2022a). Towards reconstruction of 3d shapes in a realistic environment. In *International Conference on Image Analysis and Processing*, pages 3–14. Springer.

[148] Zohaib, M., Taiana, M., Padalkar, M. G., and Del Bue, A. (2022b). 3d key-points estimation from single-view rgb images. In *International Conference on Image Analysis and Processing*, pages 27–38. Springer.

[149] Zuffi, S., Kanazawa, A., Jacobs, D. W., and Black, M. J. (2017). 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6365–6373.