Shape and Pose Estimation of Rigid Objects from a Single RGB Image

First Year Report

Florian Langer

Department of Engineering

University of Cambridge

Supervisor: Dr. Ignas Budvytis

Advisor: Prof. Roberto Cipolla

Darwin College

August 2021

Abstract

This work addresses the problem of estimating 3D shapes and poses of rigid objects from a single RGB image. With applications ranging from augmented reality to robotics and digital content creation this task is getting increasingly more attention in the research community.

Directly estimating 3D shapes and poses from a single image is a challenging problem as there exist a large variety of very different shapes that take different appearances depending on a given pose. Training networks to directly predict these shapes can to lead to unrealistic shape predictions that are either over-smoothed or artificially tessellated [Geo19; Wan+18; Gro+18; Nie+20]. Even more important [Tat+19] demonstrate that many single-view reconstruction methods do not predict shapes that match the input image, but rather predict averages of those shapes that were seen during training.

Rather than learning a full mapping from an input image to a posed 3D shape, we believe that following a hybrid approach which consists of learned modules as well as hard-coded modules capturing explicit information about the projective scene geometry can produce more accurate and consistent results. Applying this idea to the task of shape estimation we limit network predictions to the 2D image plane and use the generated predictions as 3D constraints on object shapes and poses.

Our proposed system broadly follows [Kuo+20; ISS16; GRL19] who ensure realistic shape predictions by retrieving CAD models from large scale databases [Cha+15; Sun+18] and aligning them to the objects observed in the image. However, after retrieving candidate objects [Kuo+20] attempt to directly regress object poses, leading to inaccurate predictions. Instead of predicting object poses directly we predict correspondences between a render of the retrieved CAD model and the input image. As the exact 3D coordinates of the rendered CAD model are known we can use the predicted correspondences to analytically compute object poses. We demonstrate that this procedure is more accurate than direct pose predictions.

For realistic applications object retrieval must be superseded by a deformation

step in which the retrieved CAD model is adapted to fit the observed shape. We show that the correspondences that can be used to estimate object poses can simultaneously be used to adapt the object shape. For this purpose we introduce a novel deformation procedure which stretches objects along planes in the canonical object frame. Despite its simplicity this shape adaptation procedure can cover a large range of realistic shapes. We demonstrate this through a range of experiments on Pix3d[Sun+18].

In future work we aim to incorporate more general constraints arising from interactions between objects or objects and the scene. Additionally, we want to combine the precise, sparse key-point matches with a large number of dense, but less accurate matches, in a probabilistic formulation. This will subsequently allow for more fine-grained, possibly part-based shape adaptations and more precise object pose estimates.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This report lays out the work that was undertaken this year to address the following research problem: **given a single RGB image from an indoor scene estimate the 3D shapes and poses of the rigid objects present.**

Despite continuously experiencing objects in the 3-dimensional world providing a definition of the word "shape" is difficult. For the purpose of this report a 3D shape is defined as the "external boundary of an object". Throughout this report the word pose will be used to describe the 6-DoF pose consisting of the 3D translation of the object center in camera coordinates and the 3D rotation of the shape with respect to camera coordinates. Note that the shape is one property of an object, others being for example its texture, material, density or general information commonly associated with an object (e.g. its usages or even a price range). In this report we restrict the class of objects for which shape estimation is performed to rigid objects such as tables or beds as opposed to deformable objects like humans, clothing or curtains.

1.1 Motivation

3D shape estimation from a single RGB image has numerous important applications (see Figure 1.1 for an overview).

- **Digital content creation.** Estimating shapes and poses of objects in private houses or office areas could be an important tool for furniture companies. It would enable them to provide powerful visualisations for their customers. Taking a photo of a room would be sufficient for estimating 3D models of the objects present as well as the location of the floor and the walls. In augmented

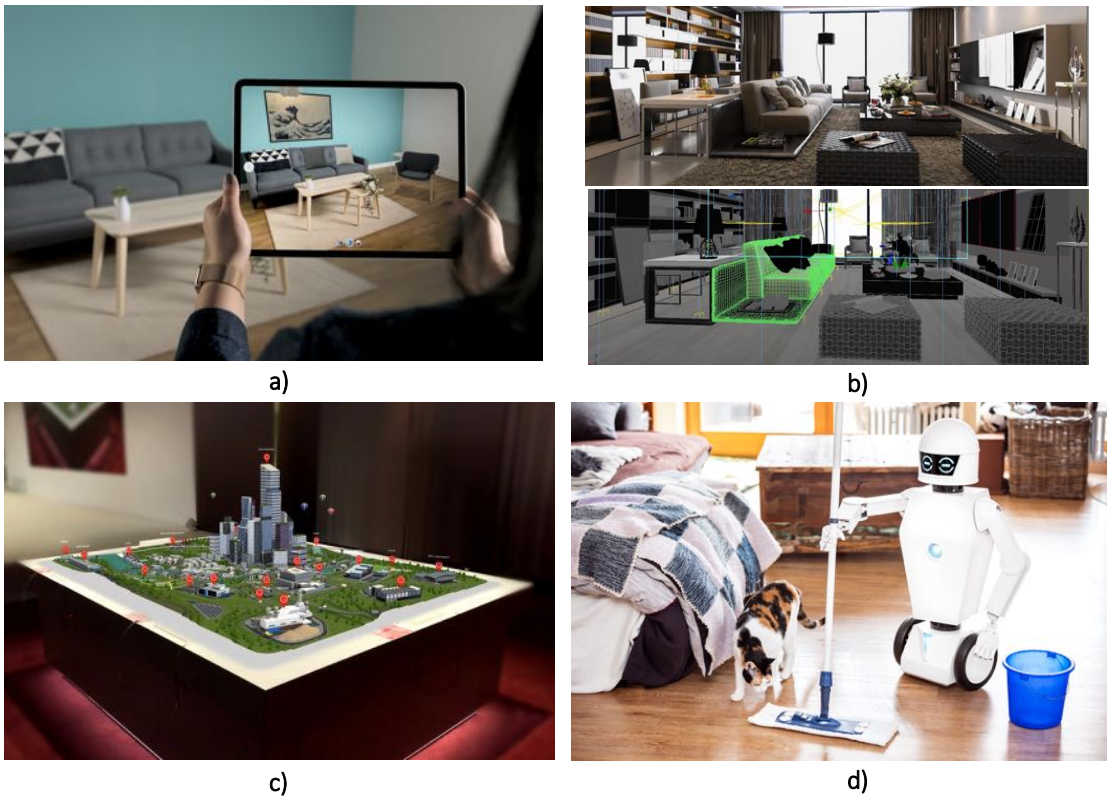


Figure 1.1: Selected applications of shape estimation from a single RGB image: a) digital content creation b) aid to 3D designers for virtual world building c) augmented reality d) robotics

reality a user could then move existing objects around, remove them entirely or see how new objects would fit into the room. Similarly, the ability to quickly create 3D models of rooms is important for online real-estate agents.

- **Aid to 3D environment designers.** A system that successfully predicts 3D shapes and poses from a single image could be an important tool for assisting 3D environment designers. Environment designers build virtual worlds in movies, computer games or for virtual reality experiences. This process is very time consuming and therefore expensive. Usually an environment designer receives images or sketches from a concept designer and then has to implement these in 3D. A system that performs shape estimation could build the scene based on these images or sketches and the scene could then be used directly or serve as a starting point for additional modifications.

- **Augmented reality.** In general, precise 3D shapes and poses of the environment are important for seamless augmented reality experiences. For example

in order to project some 3D model onto a table surface, a precise 3D model of the table is required. While additional information may be provided for example multiple views in a video or additional sensory information (e.g. LIDAR) a successful prediction from a single RGB frame will be an integral part in any robust system.

- **Robotics.** Accurate shape and pose estimation is also crucial for virtual 3D map building with applications in robotics, e.g. a cleaning robot requires accurate 3D models of its surrounding objects for tidying up.

1.2 Challenges

Estimating 3D shapes and poses from a single RGB image is difficult for multiple reasons.

- **Ill-posed Setup.** The most obvious challenge arises as a single image only provides a single viewpoint of an object. While the process of rendering a 2D image in a given 3D scene is well defined, its inverse i.e. obtaining the 3D scene from a 2D image (inverse computer graphics) has no unique solution. Humans excel at finding a plausible solution out of many theoretically possible ones by using a range of image cues, prior knowledge and intuitive 3D understanding. Humans are also particularly good at inferring information about the back side of the object (which is always missing from the image) from information present in the image and the experience of previously seen objects. Furthermore humans use expected sizes of objects and image cues to resolve the inherent scale-depth ambiguity preset in the image. So far enabling computational systems to acquire the same sort of information for predicting 3D shapes has proven to be difficult.

- **Occlusion.** When seen from a single view the front of an object can not only occlude information about its backside, but the object as a whole can also be occluded by other objects. Again this increases the difficult of accurately estimating the objects shape and pose as it reduces the amount of available information. Any realistic scene will have a certain level of occlusion present and it is therefore important that a proposed system is robust to it.

- **Large variety of shapes.** Even though rigid, man-made objects obey many regularities, such as hard edges, large planar surfaces or right angles the number of significantly different, but still realistic object shapes is very large. This makes the problem of predicting a low-dimensional parametric model such as the SMPL[Lop+15] which explains the majority of possible human body shapes in just 10 parameters[Pav+18] hard. The large variance in object shape topologies also constraints what shape representations can be used to effectively represent objects, e.g. using a popular mesh representation is difficult due to the challenges of predicting different mesh topologies (see Section 2.1.1).

- **Lack of 3D supervision.** Obtaining 3D shapes paired with real RGB images is difficult and therefore costly. Accurate 3D models either have to be scanned when an image is take or have to be modelled in a time-consuming process afterwards. Both cases take time and as such the number of available datasets that can be used for training is very limited.

- **Varying appearances.** Finally, a challenge that is always present when dealing with real images is the varying appearance of the quantity of interest, in this case the object shape. These variations in appearance due to differences in lighting, textures, materials or the presence of clutter are irrelevant for the task of predicting the object shape, but nevertheless have to be taken into account to ensure that a trained system can perform robust predictions under the present variations.

1.3 Approach

Approaches for performing shape and pose estimation from a single RGB image can be broadly classified into three different lines of research: methods relying on *direct shape prediction*, approaches that rely on *shape retrieval* and approaches that perform shape retrieval followed by *shape deformations*.

1.3.1 Research Directions

Direct Shape Prediction

For direct object shape predictions different representations are used ranging from voxels[Cho+16b], point clouds[FSG16; ZKG20], meshes[Wan+18; Geo19; Pan+19;

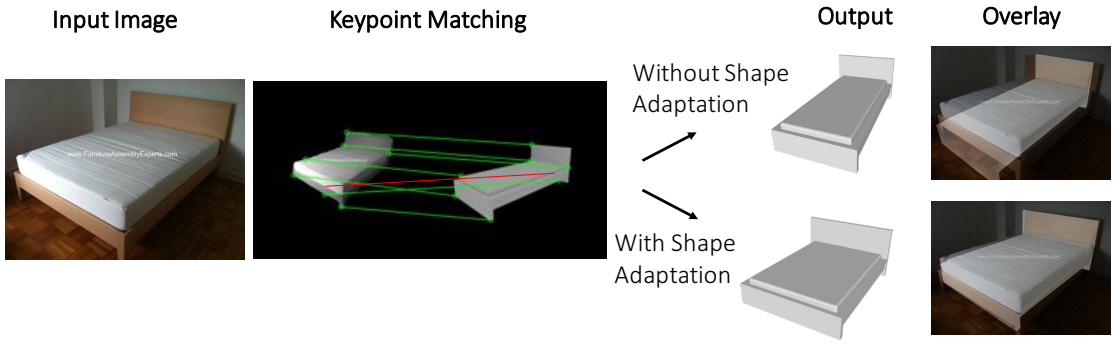


Figure 1.2: Example results. Given an input image we retrieve a CAD model rendering and perform key-point matching with the masked input image. Without shape adaptation the retrieved CAD model prediction is limited by the availability of similar CAD models in the database. When shape adaptation is performed target object shapes and their poses can be predicted very precisely.

Nie+20, packed spheres FSG16, binary space partitioning CTZ20, convex polytopes Den+20, signed distance fields Par+19 to other implicit representations Mil+20; Yu+20. Most of these approaches suffer either from a lack of precision in their predictions Cho+16b; FSG16; ZKG20; Wan+18; Geo19; Pan+19; Nie+20; Den+20 or lack applicability to a wide range of objects from different classes CTZ20; Par+19. Regardless of the representation that is chosen, directly predicting an object shape and pose from a single image is a difficult learning task as crucial information about the back side of the object is missing at both train and test time (see Section 1.2). The inherent ambiguity arising from the fact that a single view may allow for multiple possible shape predictions prevents the networks from learning effectively, causing them to perform class-averaged object shape predictions Tat+19.

Shape Retrieval

The second line of work uses existing 3D shapes not just as implicit priors that are learned during training but as an explicit shape database Cha+15; Mo+19a from which a system can retrieve shapes at test time. Retrieving multiple CAD models from the database solves the problem of predicting ambiguous object shapes as rather than predicting an averaged-shape multiple, distinct predictions can be made. Such as a set of predictions is better suited for tackling an often ambiguous problem which inherently allows multiple solutions. Retrieval-based shape estimation works especially well in man-made environments such as indoor environments which exhibit high regularities in the object shapes present. We propose a framework relying on shape retrieval in Chapter 3.

Shape Retrieval + Shape Deformation

The third line of work aims to combine the merits of both of the aforementioned approaches. In retrieving CAD models from a database it is ensured that realistic objects shapes are estimated. By subsequently deforming the retrieved shapes, the adapted shapes can fit the observed shapes more precisely compared to a pure retrieval approach. Note that shape deformations in this report imply modifying an initial shape estimate of a rigid object, but does not imply that the object itself (e.g. the table) is a deformable object (such as a human or clothing). A more detailed overview of shape deformation techniques is provided in Section 2.1.3. We propose a framework performing shape retrieval which is followed by shape deformation in Chapter 4.

1.3.2 Our Approach

The approach that we propose for estimating 3D shapes is based on shape retrieval followed by a deformation. It consists of 4 steps: (i) object detection and segmentation, (ii) CAD model retrieval, (iii) keypoint matching and (iv) pose and shape optimisation.

1. **Object detection and segmentation.** In the first step objects are detected and their instance masks are predicted.

2. **CAD model retrieval.** Similar to [ISS16; Kuo+20] a neural network is trained to embed masked RGB images and rendered CAD models into a joint embedding space. At test time given a target RGB image multiple candidate CAD models can be retrieved.

3. **Keypoint matching.** A neural network is used to match keypoints between the object in the real RGB image and the retrieved CAD model rendering [Geo+19].

4. **Pose and shape optimisation.** We modify CAD model shapes by stretching them along 3D-planes. The proposed stretching is a local operation which in contrast to a global scaling operation can modify proportions of objects within a single dimension (e.g adjusting the height of a sitting area of a chair, see Figure 1.2). Each stretch is parameterised by a plane and a stretch magnitude specifying how much an object is stretched. We optimise over the stretch

magnitudes and the object pose jointly by minimising the distance of the reprojected keypoint matches.

Our approach differs from existing work in the usage of keypoints for pose and shape estimation. We show that this is more precise than directly regressing rotation and translation parameters as done by [Kuo+20]. Currently our approach uses stretching along the three principal object axis. However, in the future predicting additional stretch planes and limiting stretch extents to variable 3D-boxes will allow for more precise deformations. Even more fine-grained shape adaptations can be accomplished by modifying object parts individually for example from dense correspondences (see Section 5.2).
We evaluate our approach on the Pix3D[Sun+18] dataset. When combining object retrieval with a geometric pose prediction we outperform existing work [Geo19; Kuo+20] on the Pix3D[Sun+18] dataset for splits containing both seen ($S1$) and unseen ($S2$) CAD models at test time. On the $S1$ split we improve over the state-of-the-art[Kuo+20] from 33.2 to **34.4** and on the $S2$ split from 8.2 to **15.2** in terms of the AP mesh score[Geo19]. We evaluate the proposed object adaptation on a range of adaptation experiments. Here we observe that dynamic fitting improves the shape predictions when no access to correct CAD models is given at test time and retrieved models have to be adapted.

1.4 Contribution

The main contributions of this work are threefold:

- We demonstrate that estimating object poses from geometric constraints is more precise than directly predicting them. We outperform leading approaches [Kuo+20; Geo19] on existing tasks.
- We introduce a novel shape deformation procedure. Under this formulation objects are stretched along the normal vectors of planes. While requiring only few parameters this procedure allows for local shape modifications which effectively capture those deformation that are required in realistic scenarios.
- We demonstrate that the proposed plane stretching formulation can be effectively used to adapt shapes from sparse correspondences. This is demonstrated on different versions of the Pix3D[Sun+18] dataset.

This report extends the work that was submitted to the *British Machine Vision Conference 2021*:

- *Langer, F. Budvytis, I. Cipolla, R.*, **Leveraging Geometry for Shape Estimation from a Single RGB Image** *In Proceedings of the British Machine Vision Conference*, September, 2020 (under review.)

1.5 Outline

The rest of this work is structured as follows. Chapter 2 presents related work on 3D shape estimation which can be categorised into methods relying on direct shape predictions, shape retrieval and shape deformations, as well as relevant datasets. Chapter 3 presents the proposed shape estimation pipeline relying on geometric constraints for pose estimation. This system is compared to existing work relying either on direct shape predictions or direct pose predictions. Chapter 4 introduces a novel plane stretching formulation which allows the adaptation of retrieved CAD model. The proposed approach is compared to a system that does not use dynamic shape and we show the need for shape adaptation in realistic scenarios. Finally, Chapter 5 consists of a concluding discussion as well as an outline of important future work.

Chapter 2

Literature Survey

This chapter presents an overview over literature related to the problem of shape estimation from single images. Section 2.1 describes and explains the various approaches from which this problem is currently tackled. Section 2.2 presents selected works from object detection and instance segmentation. Those are relevant as before a shape can be estimated the object itself has to be detected. Section 2.3 provides details of datasets that are important for the task of shape estimation. Finally, the main points of this chapter are summarised in 2.4.

2.1 Shape Estimation

Various approaches for estimating shapes of static objects from single images can be categorised into direct shape prediction methods and retrieval-based methods. A retrieval based method may subsequently apply a deformation to better fit a target shape.

2.1.1 Direct Shape Prediction

This section introduces related literature by analysing direct shape prediction methods in terms of three different criteria: the representation that was chosen, the loss function that was used and the object relations that are used.

Explicit and Implicit Shape Representations

When comparing direct shape prediction methods one property that can be analysed is whether the chosen representation is of explicit or implicit form. Explicit representations can be directly interpreted as a 3D shape.

- **Voxel.** An intuitive representation is the voxel representation under which the 3D world is discretised into cubes and object shapes are encoded as occupancy IDs of these cubes. While being easy to interpret the voxel representation suffers from an inherent trade off between accuracy and storage space due to the cubic scaling. This means that in practice the accuracy to which shapes can be estimated is limited[Cho+16b; PBF20].

- **Mesh.** Meshes [Geo19] alleviate the storage-accuracy trade-off by encapsulating information about the 2D object surface rather than its 3D volume. An object is represented as a set of vertices which are interconnected to form (usually triangular) mesh faces. The difficulty when using meshes to estimate object shapes arises due to the large variety of different object shapes with different object topologies. This is a problem as predicting a mesh from scratch (for example by predicting mesh vertices and their connectivity through edges) is too challenging. Therefore most approaches working with the mesh representation estimate object shapes by iteratively predicting vertex offsets of an initial mesh. This is a much simpler problem as it assumes a fixed connectivity of vertices. The difficulty then however remains to obtain an initial mesh with the correct object topology. Early approaches[Wan+18; Gro+18] simply deform an original spherical or ellipsoidal mesh, therefore effectively limiting its predictions to simple shapes that do not contain any holes. Other work [Geo19] estimate the initial shape in the voxel representation and use the mesh representation to refine the object boundaries. While Topology Modification Networks [Pan+19] also deform an initially spherical mesh, they iteratively update the mesh topology by removing faces which are estimated to have a large distance to the true object shape.

- **Other explicit representations.** Other representations such as pointclouds [ZKG20] or point-sets [FSG16] by predicting individual 3D points or spheres without their associated connectivity. Finally, another set of explicit representations decomposes objects into smaller shapes which can be general convex polytopes [Den+20; CTZ20] or *geometric primitives*[Tul+18].

Besides these explicit object representations other encodings exist which implicitly define an object shape. These implicit representations have to be queried to produce a 3D shape.

- **Occupancy functions.** One of the first implicit representations that was introduce are occupancy functions [Mes+19] which represents as 3D shape as the continuous decision boundary of a classifier. Mathematically the occupancy function that is needed for this is defined by $o : R^3 \rightarrow \{0, 1\}$ and [Mes+19] show that a neural network can be trained to apporximate o well.

- **Signed distance fields.** Very related to occupancy functions are signed distance fields. They can be understood as a more general scalar-field $sdf : R^3 \rightarrow \{-\infty, \infty\}$ which does not just contain the information whether a given point resides inside (-) or outside (+) of the object but also its distance to the object surface. Again [Par+19] show how to effectively learn this function for classes of objects.

- **Neural radiance fields.** Recently [Mil+20] showed that neural networks can not only be trained to implicitly encode 3D shapes, but entire scenes including information about textures and lighting. For this purpose in addition to the 3D scene coordinate $\mathbf{x} \in \mathcal{R}^3$ they provide information about the viewing direction $\mathbf{d} \in \mathcal{R}^2$ and regress the color $\mathbf{c} \in \mathcal{R}^3$ and scene density $\sigma \in \mathcal{R}$, $NeRF : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$. Despite showing very accurate results in terms of novel view synthesis this original formulation of neural radiance fields could not be used for object shape estimation as every scene has to be individually optimised for. [Yu+20] extend the original formulation by conditioning the scene network on features of an input image. In this way general scene priors can be learned during training which enable inference at test time from just a single image. A great advantage of using neural radiance fields is that they do not require 3D models as supervision during training, but can just use a collection of images with known relative camera poses. This is particularly important as the lack of 3D supervision was identified as one of the great challenges in Section 1.2.

Loss function

Another dimension along which direct shape prediction methods can be analysed is in terms of the loss function they are using. This is important as the loss function that is used has a great influence on the training process and the final performance that can be achieved. Setting up a problem with an unsuitable loss function may prevent a network from learning effectively. In general we distinguish between clas-

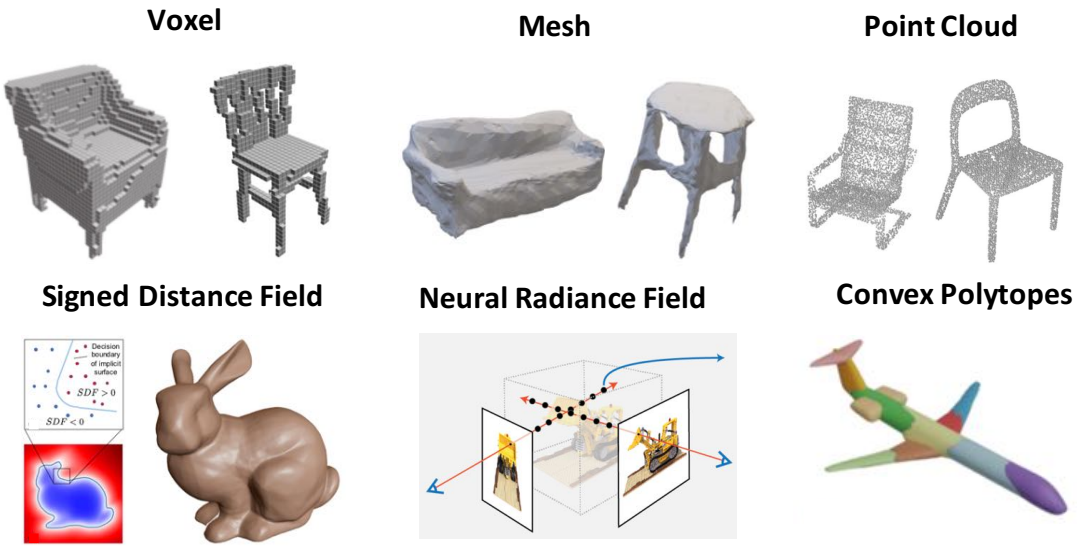


Figure 2.1: Direct shape prediction methods employ a range of different shape representations. From left to right and top to bottom: Voxel[Cho+16b], Mesh[Geo19], Point Cloud[ZKG20], Signed Distance Field[Par+19], Neural Radiance Field[Mil+20] and Convex Polytope[Den+20].

sification losses and regression losses. This distinction is important as numerous works have shown that networks that perform classification are more robust than those performing regression. Analysing different direct shape prediction methods in terms of the losses used we note that methods relying on the voxel representation[Cho+16b] and on occupancy functions[Mes+19] naturally perform a classification task (i.e. classifying whether a given point or voxel is inside or outside of the object). This is in contrast to approaches based on meshes or point-clouds which directly regress points on the object surface. The most important loss function used in this context is the Chamfer-distance. It is used in many works including [Geo19; Pan+19; Gro+18] and compares two shapes by sampling point-clouds P and Q from them and computing their distance based on

$$\mathcal{L}_{\text{cham}}(P,Q) = |P|^{-1} \sum_{(p,q)\in\Lambda_{P,Q}} \|p-q\|^2 + |Q|^{-1} \sum_{(q,p)\in\Lambda_{Q,P}} \|q-p\|^2 \qquad (2.1)$$

where $\Lambda_{P,Q} = \{(p, \arg\min_q \|p-q\|) : p \in P\}$ is the set of pairs (p,q) such that q is the nearest neighbor of p in Q.

Signed distance fields[Par+19] regress the distance to the object surface and minimise the L1 distance of the prediction. Other methods that decompose object

into shape elements need to regress parameters of bounding planes[CTZ20; Den+20] or shape parameters[Tul+18] but also compute the loss for the final 3D shapes based on concepts similar to the Chamfer-distance, e.g. by penalising if the predicted shape is not within the ground truth shape and vice-versa[Tul+18].

Finally, approaches using neural radiance fields differ from previously discussed loss functions as they are applied on 2D images as opposed to 3D shapes. This changes the learning problem significantly and as previously mentioned allows for training without explicit supervision.

Single Object vs Multi-Object

A third dimension along which direct shape prediction methods can be analysed is whether they perform single object predictions or take other object or even the whole scene into account when making predictions. Among the systems analysed above [Wan+18; Pan+19; FSG16; Tul+18; Cho+16b] perform single object predictions. To emphasize systems that can predict multiple object shapes from a single image, but do so by making independent predictions for each of those objects are also considered single object reconstruction methods here[Geo19]. For these methods occlusion which is present in almost all realistic scenes is a great problem as the amount of available information based on which a prediction can be made is reduced. This is in slight contrast to methods [Nie+20; PBF20; Yu+20] that predict object shapes jointly. While for these occlusion can still be an issue, it can also provide information about relative positioning of objects with respect to each other which can place constraints both on the estimated poses and shapes. Approaches based on neural radiance fields[Yu+20] naturally model occlusion in the estimated scene density function. Additionally some approaches explicitly use physicality constraints, for example by imposing that solid objects can not intersect by imposing a collision loss[PBF20]. Other work[Nie+20] try to take interior design principles into account in their pose predictions by modelling object relations as an attention sum which is provided as input for the final pose prediction.

2.1.2 Shape Retrieval

In contrast to direct shape prediction methods another line of work estimates 3D shapes by retrieving CAD models from a database and aligning them to the objects observed in an image. This guarantees predicting valid shapes that do not suffer from artificial tesselation or oversmoothed boundaries. However, this comes at the

cost of flexibility at test time as now the range of shapes that can be estimated is limited by those present in the dataset. While one could argue that direct shape prediction methods are also limited in their shape predictions by those seen in the training data, there is usually the hope that neural networks can perform some amount of interpolation between those shapes at test time.

Embedding Space

An important aspect along which shape retrieval methods can be analysed is by the embedding space they use. However, before comparing methods it has to be noted that retrieving objects from an embedding space is just one approach for choosing one of N discrete objects. Perhaps a more obvious approach is to perform a classification over N objects (as done by [Eng+21]). Nevertheless, in practice almost all approaches[Aub+14; Li+15; ISS16; GRL19; Kuo+20; Man+20] rely on an embedding space as it provides a more natural encoding of similarity than classification does. This is important as a given object may be well approximated by a set of different shapes from the database rather than having one perfect match. Additionally, an embedding space provides the possibility of adding further objects to the database after training which can then be retrieved at test time. This is not possible when performing classification.

Before constructing the embedding space methods relying on shape retrieval have to deal with the differences in modality between the 2D input images and the 3D shapes they hope to retrieve. Most approaches [Aub+14; Li+15; ISS16; Kuo+20; Man+20] solve this problem by following the idea of [Che+03] in representing a 3D shape as a collection of 2D images of that shape rendered from evenly sampled viewpoints. An alternative approach[GRL19] maps input images and 3D shapes to an intermediate representation called Location Fields. Location Fields have the same shape as images but instead of color channels provide the 3D scene coordinate in the canonical object frame for every pixel. As they only encoded information about the 3D geometry of the object they are invariant to changes in lighting or texture of an object. More importantly, and unlike other conceivable representations, such as surface normals or depth, a certain 3D point on the object is always encoded by the same 3D coordinates. This is different to surface normals or depth where the encoding of a 3D point on the object depends on the viewpoint of the camera. While this in-variance may simplify the networks task of learning this representa-

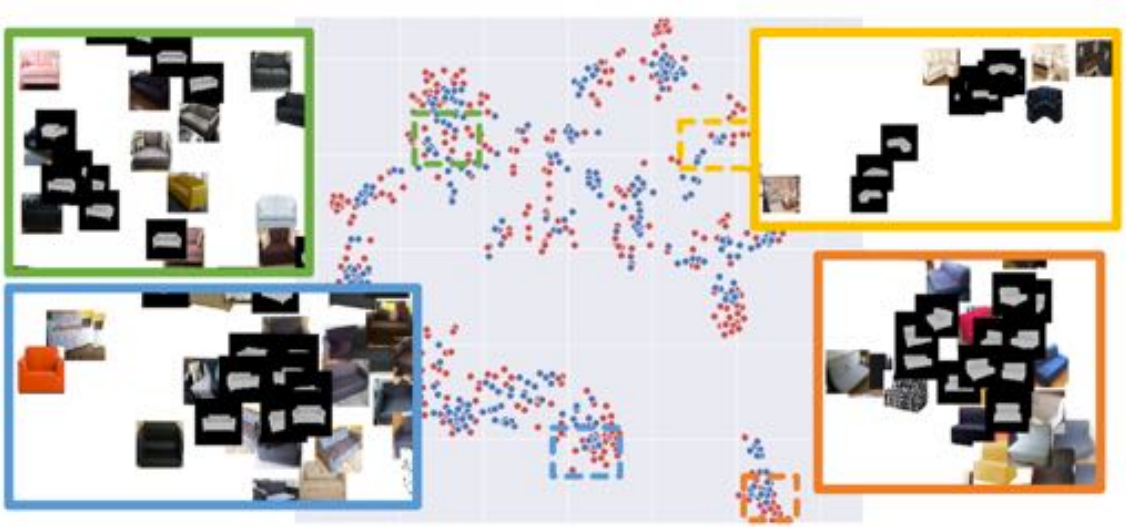


Figure 2.2: Learned joined embedding space of CAD model renderings and masked RGB images of Mask2CAD[Kuo+20] (visualised using t-SNE[MH08]). Similar CAD models renders and real images are embedded close to each other. Comparable embedding spaces are also learned by other retrieval approaches[ISS16; GRL19]. Figure from [Kuo+20].

tion, predicting Location fields accurately remains a challenging problem[GRL19]. Focusing now on those approaches that use rendered images to represent 3D objects we find that early work[Aub+14; Li+15] on shape retrieval uses hard-coded features of the Histogram-of-Gradients[DT05] descriptor to construct the embedding space. These are then replaced by features learned by convolutional neural networks in [ISS16; Kuo+20]. Both [ISS16; Kuo+20] minimise the cosine distance of extracted features. [Kuo+20] in particular use a variation of the triplet loss[Che+10]. This minimises distances between embeddings of an image and the corresponding CAD model rendering and maximises distances to non-corresponding CAD model renderings. For selecting positive and negative examples they perform hard-example mining. However, it is unclear if performing hard example mining is beneficial in this case as it forces the network to match images with very different features (see Section 3.2.2).

Pose Estimation

After retrieving a 3D shape from a database the next essential step to perform 3D shape estimation is to align the retrieved shape with the observed object in the image. Depending on the problem setup this can either be the 6-DoF pose consisting of an object orientation and a translation or a 9-DoF pose which additionally con-

tain a three dimensional object scaling. As opposed to shape retrieval where most existing works follow a similar approach, there exist a range of different techniques for estimating object poses. Probably most intuitive when using neural networks is to directly regress the required parameters as done by [Eng+21]. [Eng+21] predict the object orientation by regressing a 9-dimensional output which is interpreted as a matrix and following [Lev+20] projected into $SO(3)$ to obtain a valid rotation matrix. However, accurately regressing object poses directly is difficult. Particularly, predicting the scale as well as the object position along the optical axis is challenging due to the inherent scale-depth ambiguity. Mask2CAD[Kuo+20] avoid this problem by limiting predictions to 6-DoF poses without scale and using the ground truth z value in their test time predictions. They estimate object translations by regressing the offset of the reprojected 3D object center into the image plane from the 2D bounding box center and then use the ground truth z to reproject the object center into 3D space. For the rotation they first perform a classification into one of 16 rotation bins and then regress the offset to the true rotation using quarternions. In general it is difficult to achieve precise object poses with such direct prediction methods as it is hard for neural networks to make accurate predictions in 3D space. Another approach is to establish correspondences between image pixels and 3D world coordinates in the object canonical frame. This is essentially provided by the Location Field descriptor[GRL19] (as mentioned in Section 2.1.2) which estimate such a correspondence for every input pixel. Once these correspondences are established an absolute pose estimation algorithm such as PnP[Gao+03] can be used to compute the object pose. Given precise correspondences the pose can be computed exactly. However, the difficulty with this approach lies in accurately predicting the Location Fields[GRL19].

Another line of work iteratively refines a pose by comparing the object rendered under the current pose to the input image. When comparing real images with rendered CAD models this comparison is best done in feature space which is less sensitive to differences in lighting and object textures compared to RGB space. For this purpose Im2Cad[ISS16] compare a set of outputs from various convolutional layers of the VGG[SZ15] network and use a derivative-free optimizer[Pow94] for the optimisation. [Gra+20] go further by establishing dense correspondences between image features and demonstrating how associated gradients can be directly propagated to mesh vertices for efficient pose refinements. While more computationally intense these methods allow for more precise object poses than direct prediction methods. Finally, going beyond single image predictions [Man+20] show how accurate pose

estimates can be obtained from videos by combining single image predictions in a globally consistent, multi-view constraint optimization.

Single Object vs. Multi Object

Just as for direct shape prediction methods retrieval based methods can be analysed in the extent to which they perform joint object predictions. Some approaches [Kuo+20; GRL19] perform entirely independent object predictions. Others such as Im2CAD [ISS16] following the render-and-compare approach naturally use object occlusion for informing pose updates. Again there exists work [Eng+21] which employs a collision loss to penalise overlapping object predictions (similar to [PBF20]). Another interesting work [Ave+20] creates a fully-connected scene graph where nodes represent objects, the floor and wall segments. [Ave+20] performs message-passing to accumulate information about object-object relations and object-layout relations. These are then used to predict support relations and relative object orientations to achieve globally consistent configurations. Note that while the input for [Ave+20] are 3D scans as opposed to a single RGB images some of the insights and methods can be transferred to single image reconstruction methods.

2.1.3 Shape Deformation

Shape deformation or shape adaptation is the process of modifying an initial 3D shape estimate into a target shape. In this section existing work on shape deformation is broadly classified into methods relying on control points, methods attempting to learn a more general shape deformation space resulting in per-vertex deformations and shape deformation methods that take a retrieval step into account.

Deformations Based on Control Points

Rather than directly modifying mesh vertices a range of approaches modifies a smaller set of control points which in turn define deformations of all mesh points. Early work [Jac+18; Kur+18] on shape deformation relied mainly on Free Form Deformation (FFD) which (different to the name implies) rely on a set of control points laying on a 3D grid. Formally FFD can bed defined as follows: Given an initial point $p = (u, v, w)$ and control point offset Δ_{ijk} at $p_{ijk} = (i, j, k)$ the deformed

point p' is defined as

$$p' = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} (p_{ijk} + \Delta_{ijk})\, B_{l,i}(u) B_{m,j}(v) B_{n,k}(w)\,. \tag{2.2}$$

Here $B_{n,m}(x) = \begin{pmatrix} n \\ m \end{pmatrix} (1-x)^{n-m} x^m$ is a binomial function and l, m, n are the sizes of the control point grid. The binomial functions control the influence of the control points for a given point p and ensure that p is affected more by nearby control points. More recent work[Yif+20] uses control points which do not lie on a regular 3D grid, but rather are the vertices of a cage which can be thought of as a coarse mesh encompassing the shape that is to be deformed. Cage deformations are built on top of the idea that points $p \in \mathcal{R}^3$ can be represented as the weighted sum of a set of cage vertices $\mathbf{v}_j$

$$\mathbf{p} = \sum \phi_j^{\mathcal{C}}(\mathbf{p}) \mathbf{v}_j \tag{2.3}$$

where the weight function $\left\{\phi_j^{\mathcal{C}}\right\}$ depend on the relative position of p with respect to the cage vertices $\{\mathbf{v}_j\}$. Given a cage deformation transforming cage vertices to v' the transformed point p' is given by the weighted sum of transformed cage vertices v' with previously computed weights $\phi_j^{\mathcal{C}}(\mathbf{p})$ $\mathbf{p}' = \sum \phi_j^{\mathcal{C}}(\mathbf{p}) \mathbf{v}'_j$.

This is perhaps a more intuitive arrangement of control points as it is unclear why control points are needed within a a shape or far from its surface. Additionally, using a cage deformation allows to adapt the deformation resolution (as opposed to laying on a fixed gird) allowing the procedure to adapt to specific shape categories and source shapes. This has shown to produce deformations that can preserve details very well (see Figure 2.3). Recently, [Jak+20] demonstrated that cage based deformation can be effectively controlled by just a few keypoints (on the order of 10) enabling even more intuitive user-based shape manipulation. [Jak+20] learn those keypoints in an unsupervised way by deforming a source shape into a target shape by adjusting the position of source keypoint to match target keypoints. By imposing a loss based on the Chamfer-distance between the deformed and the target object, they learn consistent and semantically meaningful keypoints across different objects of the same category. They also learn how these keypoint displacements effectively translate to cage vertex offsets which allow controlling shape the shape deformation. The sparsity of keypoints needed makes this approach very promising for deforming a retrieved object when the input signal is just a single RGB image.

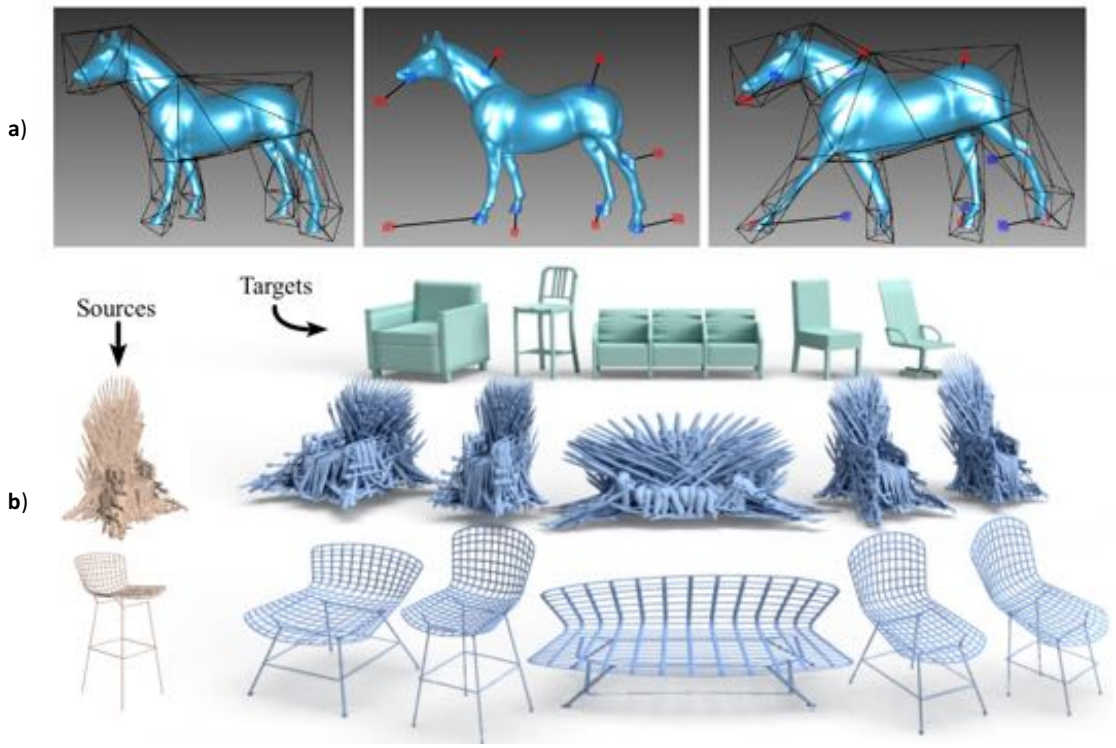


Figure 2.3: a) Visualisation of the cage deformation (Figure from [SF10]). An initial shape is encompassed by a coarse cage mesh (left). The shape can be deformed by adjusting vertices of the cage (middle). The deformed shape preserves the structure and details of the original shape (right). b) Results obtained by [Yif+20] when deforming a source mesh to fit a target by predicting cage vertices offsets. Figure from [Yif+20].

Vertex-based deformations

Unlike previous methods which rely on a set of control points object shapes can also be deformed by predicting individual mesh vertex offsets. Existing work [Gro+19] approaches this by encoding both source and target shape into a latent representation which are used in conjunction to predict parameters which are used in a deformation network. This deformation network takes as input a 3D point on the source mesh and learns to directly predict the corresponding point on the deformed mesh, based on the predicted parameters and additionally learned parameters. Interesting about their approach is that they enforce a cycle-consistency loss by imposing that the shape one ends up with after deforming shape A into B and then deforming the resultant shape back into A is as similar to A as possible (as measured by the Chamfer-distance). Another approach aims to learn a shape embedding space where vectors in this embedding space (for example obtained by subtracting two shape embeddings from each other) correspond to valid deformation. This is in

contrast to other approaches such as [Gro+19] which concatenate latent representations of shapes and feed these into an additional MLP to obtain a deformation encoding. Ideally [Sun+20] want to allow deformation transfer such that if ab is the transformation from shape A to shape B, this vector can be shifted and added to the encoding of shape C to get a valid new shape. But this is difficult to do directly as the possible deformations depend on the source shape A. [Sun+20] solve this by introducing learned source-dependent shape dictionaries which contain information how a shape deformation affects given points on the source shape. Rather than treating the difference of encodings of shape A and B directly as a deformation they multiply it with the shape dictionary of the encoding of C to obtain the final deformation. Finally, [Jia+20] model the deformations as a flow process via a flow function $\boldsymbol{f}_\theta(x(t), t)$ where intermediate deformations are controlled by an interpolation parameter $t \in [0, 1]$. They show that such deformation flows can be learned well with neural network and help to disentangle global features such as shape topology and connectivity from more local features controlling for example the dimensions of individual parts and their position.

Combining Deformation and Retrieval

Another class of approaches which is particularly suited for object shape estimation rather than general shape understanding deformation combines both object retrieval deformation and deformation in a single pipeline. Notable work[Uy+20] demonstrates that retrieving an object solely based on the similarity to the object in an input image may lead to sub-optimal behaviour if the retrieval is followed by a deformation step. Intuitively this makes sense as the initially most similar object may not be the one that can best be deformed to fit a target. They construct a deformation aware embedding space by introducing egocentric distance fields which are predicted for all source shapes and encapsulate information about how well the source shape can be deformed to match other shapes in its neighbourhood. When estimating a target shape t they retrieve a source s which has minimum distance under the estimated distance field of s rather than which minimises the euclidean or cosine distance. In [Uy+21] the authors go even further by learning not just a deformation aware embedding space but learning deformation and retrieval jointly. This has the additional benefit that their learned deformation procedure can focus on realistic retrieval scenarios which will be useful at test time (as opposed to wasting capacity on learning deformations which are unlikely under the learned retrieval pro-

cedure). The deformation procedure they introduce is novel as it allows part-based object deformations[1]. For each part of a source shape they predict three dimensional scaling and translation. The predicted transformations are then mapped onto the space of connectivity preserving transformations to yield the final shape deformation. While currently this approach has not been tested on real images in-the-wild, part-based deformations will play an important role in true 3D shape understanding and deformation.

2.2 Object Detection and Segmentation

In order to precisely estimate object shapes from RGB images the objects of interest first have to be detected. This detection can be in the form of a 2D bounding box or additionally providing a pixel-wise instance mask. While not strictly necessary accurate instance masks are often crucial for precise shape estimation. This section presents a brief overview of existing approaches for object detection and segmentation. It is structured by splitting existing work into two-stage methods, known as region-based or proposal-based methods, single stage methods, also called proposal-free or anchor-free and methods that build on the Transformer architecture[Vas+17].

2.2.1 Two-Stage Methods

The first example of a two-stage, region-based method for object detection is R-CNN[Gir+13] (the R stands for region). In a first stage [Gir+13] generate a large number of region-proposals (of the order of 2000 per image). In the second stage they create feature representations of the proposed regions using a CNN. Based on these features they subsequently classify the proposed region and perform non-maximal suppression to only keep the highest scoring bounding box per object. This showed superior performance to existing detection methods which were mainly based on handcrafted HoG-like features. In [Ren+15] speed up region-proposal generation by training a neural network for the task rather than performing a selective search[Uij+13] as was used in [Gir+13]. Finally, [He+18] extended Faster-RCNN[Ren+15] with an instance segmentation branch. Mask-RCNN[He+18] does this in an intuitive way, i.e. by extracting features from the region proposal and applying convolutions. However, they show that for accurate instance segmentations they require precise features which they obtain by bi-linearly interpolating origi-

[1] They obtain part-level annotations from the PartNet[Mo+19a] dataset.

nal image features for the new grid in the region proposal. Mask-RCNN [He+18] is widely used today and still constitutes a strong baseline for more complex architectures. Inspired by rendering techniques PointRend [Kir+20] perform adaptive sampling of query pixels for which a label is to be predicted. This adaptive sampling which relies on an iterative sub-division strategy allows them to sample high frequency regions such as object edges very densely leading to more accurate edges and the reconstruction of finer details.

2.2.2 Single-Stage Methods

In contrast to the two-stage approaches above where a region proposal is followed by a classification of that region there exist a range of works [Red+16; Dua+19; BNG17; Nev+19] which perform object detection (and some also instance segmentation) in a single-stage. [Red+16] approaches this by mapping an input image onto a 7×7 grid and directly predicting bounding boxes and class labels from each grid cell. While being extremely fast, the model is limited by the number of nearby objects it can predict as each cell can only predict two bounding boxes. An alternative one-stage detector is CenterNet [Dua+19] which is a keypoint based approach built on the ideas CornerNet [LD19]. CornerNet [LD19] first introduced keypoints from which to recognise objects which approach to alleviate the need of anchor boxes. However, CornerNet [LD19] only recognise object based on a top-left and bottom-right estimated keypoint and CenterNet [Dua+19] show that by additionally predicting a center keypoint the network has access to more information regarding the object leading to more accurate detections. Another line of one stage approaches [BNG17] learns per-pixel embeddings by training on a discriminative loss function. This allows to segment object based on their similarity in the embedding space and is particularly suited for segmenting objects that were unseen during training as the learned embedding can still be robust to variations in shape.

2.2.3 Transformer-Based Methods

In natural language processing the architecture of choice over the last few years has have been Transformers as introduced by [Vas+17]. In [Dos+21] the authors tried to adapt the original text sequence-based Transformer architecture to images with as few modifications as possible. For this they split the original image into a 16×16 grid of small patches and embed these patches individually with a learned linear projection. Adding learned positional encodings the patch embeddings are fed into

the Transformer Encoder which consists of a series of multi-head attention layers followed by regular MLPs. While [Dos+21] show good results on image classification the low-resolution features extracted at a constant scale makes it unsuitable for dense vision tasks such as segmentation. Recently [Liu+21] alleviate these problems by extracting features on hierarchy of different scales. Starting by extracting very small, high resolution features corresponding to patches of 2×2 pixel, they iteratively merge embeddings of neighbouring patches. While doing so they compute multi-head self-attention. However, rather than computing this globally i.e. for all pairs of patches, they restrict attention to a sliding window. This restricts the computation of attention to local regions of non-overlapping windows and by including a shifted window they effectively allow for cross-window connections. When applying Transformers on images restricting attention to local regions of different scales is valuable. The reason for this is that it better encodes the notion of locality intrinsic to images compared to simply learning positional encodings as previously done by [Dos+21]. [Liu+21] showed state-of-the-art performance on the COCO[Lin+15] and ADE20K[Zho+17] dataset and can serve as an alternative backbone compared to convolutional neural networks.

2.3 Datasets

This section outlines existing datasets that are crucial for training neural networks to predict object shapes from single images. The lack of real, large-scale datasets highlighted in Section 2.3.1 motivates the usage of synthetic datasets presented in Section 2.3.2.

2.3.1 Real Datasets

Creating accurate, large-scale datasets for the task of predicting 3D shapes from single RGB images is very costly. One challenge is that unlike for other tasks such as object recognition or *instance segmentation* annotation data can not be created independently after the data collection process. This means that vast amounts of available RGB images of indoor scenes can not be used as no geometric information about corresponding shapes is available. At the same time ever increasing online CAD model databases by itself are not useful as they do not come with corresponding RGB images of their objects in realistic settings. Creating pairs of RGB images and precisely aligned CAD models requires access to either the space in which the

Figure 2.4: Point1 - Pix3D precise alignments and shapes, still fairly small size many images catalogue images (ca. 20 % check) Point 2 - ScanNet realistic environments, imprecise alignments, useless for learning correspondences

RGB image was taken such that 3D scans can be performed of the objects present or direct access to the correct CAD models. One early dataset [LPT13] was created by aligning available IKEA CAD models with corresponding in-the-wild product images. However, this small-sized dataset only consists of 90 3D models and 759 images. It was expanded to form the Pix3D dataset [Sun+18] by increasing the number of 3D CAD models (through online search as well as object scanning) to 395 and the number of images to 10,069. While the object-image alignment for Pix3D is very precise, some images (ca. 20 % are either catalogue or posed images where the object was placed in the centre of the room and is clearly visible in the middle of the frame. Predicting shapes for these images is slightly easier compared to more realistic settings such that a system trained on this dataset might be less accurate in the real world.

Another dataset is Scan2CAD[Ave+19] which is built on top of the ScanNet dataset[Dai+17]. It provides CAD model annotations for 14225 objects in 1506 scans. With more than 2.5 million images of these scenes it provides orders of magnitudes more training pairs. However, its major drawback is that matched CAD models are obtained from the ShapeNet dataset [Cha+15] and do not always correspond accurately to the objects in the scene. This makes learning precise object shapes and poses very difficult.

While other datasets with 3D annotations exist, most notably NYU-D[NF12], SUN RGB-D[SLX15], 2D-3D Stanford[Arm+17] or Matterport 3d[Cha+17a] these only contain 3D surfaces (either approximated as partial point-clouds or partial meshes) as opposed to full 3D shapes.

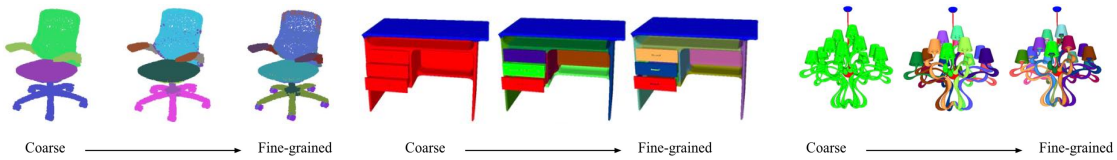


Figure 2.5: Visualisation of part level annotations of PartNet[Mo+19a]. Figure from [Mo+19a].

Figure 2.6: RGB images rendered from SceneNet RGB-D are realistic in terms of objects, textures and lighting, but unrealistic in their object room layouts.

2.3.2 Synthetic Datasets

The small dataset sizes or lack of accurate 3D shape and 2D image correspondences highlighted in Section 2.3.1 increases the importance of synthetic datasets. In general a crucial resource of 3D shapes is the ShapeNet [Cha+15] repository containing over 50,000 3D CAD models. While these CAD models by themselves can not be directly used for training networks to estimate shapes many existing works [Xie+19; Wan+18; Mes+19; Cho+16a; Geo19] pre-train their systems by rendering the CAD models in front of a white or randomly sampled background. A subset of ShapenNet[Cha+15] CAD models was annotated to include part labels in PartNet[Mo+19a]. Going beyond ShapeNet[Cha+15] this dataset allows learning object part-structures which is gaining increasing research interest [Uy+21] as it is an important step towards true 3D shape understanding.

Finally, there exist datasets with synthetically rendered scenes, most importantly SceneNet RGBD[McC+17] from which 2D image to 3D object shape correspondences can be learned directly. SceneNet RGBD[McC+17] provides 5 million photorealistic renderings of 16,895 room setups containing objects randomly sampled from ShapeNet[Cha+15]. The synthetic nature of the dataset allows for perfect 3D shape alignment with the 2D images. Despite realistic rendering quality and object variety its main drawback are the unrealistic room setups. As room configurations are achieved by randomly dropping objects from the ceiling many sampled scenes

appear highly unrealistic. This makes learning and exploiting object-object and object-room dependencies which are present in the real world impossible. Nevertheless, this synthetic dataset (and others) may still prove to be crucial for the task of 3D shape estimation due to the almost endless, semi-realistic amount of training data they can provide.

2.4 Summary

This chapter served as an overview over literature related to the task of 3D shape estimation from a single RGB image. Existing work can be broadly categorised into direct shape predictions, retrieval-based methods and deformation-based methods. While direct shape predictions are difficult to perform accurately, purely retrieval-based methods are intrinsically limited by the range of available CAD models. Deforming retrieved CAD models overcomes this limitation and can provide precise predictions. Currently, only few datasets exist from which shape predictions can be learned directly due to the difficulty and associated cost in collecting accurate training data. This makes the usage of synthetic datasets crucial for training more robust and accurate systems.

Chapter 3

Shape Estimation via Object Retrieval

This chapter presents our approach towards obtaining precise object shapes from RGB images. Unlike other retrieval-based shape estimation [Kuo+20] (see Section 3.1 for an overview of their approach) we do not directly predict object poses, but use keypoint matches to introduce geometric constraints from which the object pose can be computed. Section 3.2 explains in detail the system that was developed and the reasons for the design choices that were made. Section 3.3 provides information about the experimental setup, including information about the dataset that was used, the evaluation metric employed and the hyper-parameters that were set. We present experimental results in comparison to existing systems in Section 3.4. Finally, limitations of the proposed system are highlighted in Section 3.5 which is followed by a brief summary in Section 3.6.

3.1 Related Work

The key-competitors for performing shape and pose estimation from a single RGB image is the direct prediction method Mesh-RCNN [Geo19] and the retrieval-based method Mask2CAD [Kuo+20].

- **Mesh-RCNN**. Mesh-RCNN [Geo19] performs objection detection and segmentation with the popular detection framework Mask-RCNN [He+18]. In addition to the existing bounding box prediction branch and instance segmentation branch it trains a new voxel branch and mesh refinement branch. The voxel branch is used to generate an initial estimate of the objects shape in

the voxel representation. Its main purpose is predicting an object of the correct topology which is subsequently refined by transforming the object into the mesh representation and iteratively adjusting individual mesh vertex positions. The mesh is treated as a graph where mesh vertices correspond to nodes and mesh edges become edges in the graph. In order to obtain information for predicting vertex displacements graph convolutions are performed which accumulate information over the local neighbourhood of the vertex. During this process alignment is maintained between image features and vertices by reprojecting vertices into the image plane and performing bi-linear interpolation of the image features.

- **Mask2CAD.** Mask2CAD is a retrieval-based method which learns a joint embedding space of CAD model renders and real images. Using ShapeMask [Kuo+19] as an object detection and segmentation framework, they crop the features of a ResNet [He+16] backbone encoder based on the predicted segmentation masks. The cropped features are used as an input to a sequence of convolutional layers which output a 128 dimensional feature vector through average pooling in the last layer. A similar architecture (without cropping) is used to encode renderings of CAD model. At train time [Kuo+20] minimise the cosine distance in feature space between masked RGB images and corresponding CAD model renderings using a noise contrastive estimation loss [OLV18]. This minimises the distance of the encoding of the real RGB image to the encoding of the rendered CAD model if the two object are the same and maximises it if they are different. Mask2CAD [Kuo+20] employ hard example mining to sample CAD model renderings from which the network can learn efficiently. At test time they embed the masked RGB image and retrieve the CAD model corresponding to the nearest neighbour rendering. In order to predict the object orientation Mask2CAD [Kuo+20] first perform a classification over 16 rotation bins classifying the rotation around the vertical and then directly regress the 3DoF rotation offset. For predicting the object translation [Kuo+20] regress the difference of the 3D object when reprojected into the image plane to the 2D bounding box center. This provides them with the pixel bearing pointing towards the 3D object center. Multiplying this with the ground truth z translation (which they require at both train and test time) allows them to estimate the object translation.

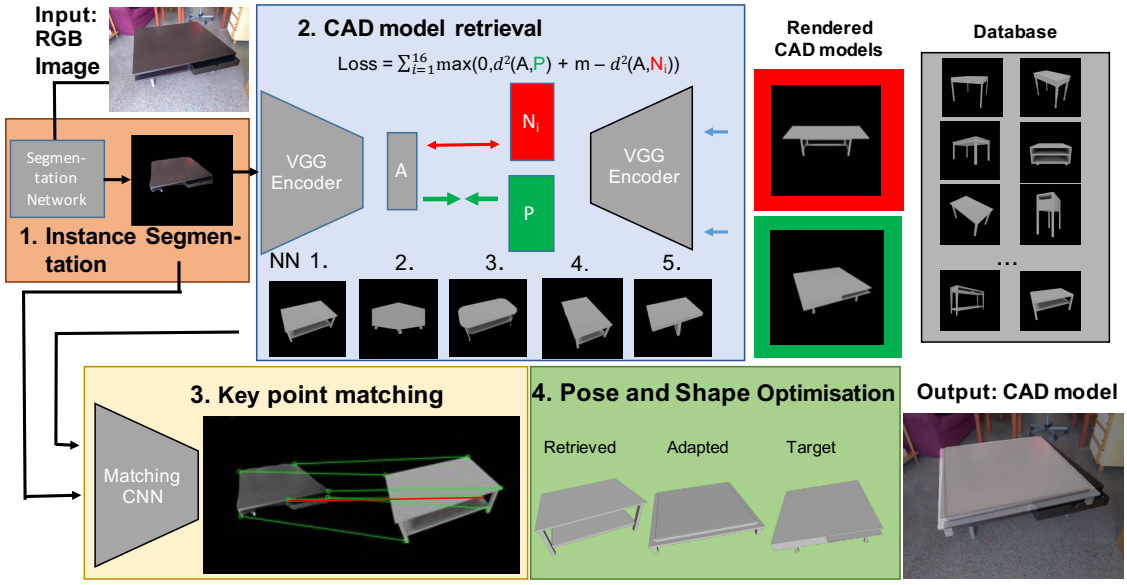


Figure 3.1: Method: Given an RGB image we perform object detection and instance segmentation (step 1). We retrieve the nearest neighbour CAD model renderings (step 2) and perform keypoint matching (step 3). The keypoint matches are subsequently used to jointly optimise over the shape and pose of the object (step 4).

3.2 Approach

Our method consists of four steps: (i) object detection and instance segmentation, (ii) CAD model retrieval, (iii) keypoint matching and finally, (iv) pose optimisation.

3.2.1 Object Detection and Instance Segmentation

For object detection and instance segmentation we train a Swin-Transformer [Liu+21] network on the Pix3D [Sun+18] dataset. During training we employ standard image augmentation techniques including random crops, scaling, rotations, horizontal flipping and random brightness and contrast adjustments. We also report results on segmentation masks obtained using Mask R-CNN [He+18] trained by [Geo19]. We do this as we found our approach to be very sensitive to the segmentation predictions. While our trained Swin transformer [Liu+21] network is sometimes able to produce more accurate segmentation masks of fine grained object, the overall performance in terms of the AP mask score is similar (see Figure 3.2 for qualitative comparisons).

Figure 3.2: Comparison of predictions obtained using Mask-RCNN [He+18] (left) and a Swin Transformer [Liu+21] (right). While overall predictions are of comparable quality, the Swin-Transformer [Liu+21] is occasionally able to generate more fine-grained predictions (see e.g. row 3 for both the S1 and S2 splits. Explanations of the two splits are provided in Section 3.3.1). Note also how both approaches struggle to predict segmentation masks for objects of unseen shapes in the S2 split (e.g. the table in row 4, the wardrobe in row 5 or the bookshelf in row 6).

3.2.2 Learning a Joint Embedding Space

Inspired by [Aub+14] we retrieve CAD models based on the visual similarity of their renderings to an input image. For this purpose we render a given CAD model in regular intervals in its orientation. This step is important as it bridges the domain gap from CAD models to RGB images and increases the similarity of the two inputs therefore simplifying the matching task. We use a single VGG [SZ15] encoder for encoding both real masked RGB images and rendered inputs. This encoder is trained on a triplet-loss[Che+10] where given an anchor RGB image A the rendering of the corresponding CAD model in the most similar orientation is used as a positive example P and selected renderings of differing CAD models are used as negative examples N_i,

$$\mathcal{L} = \sum_{i=1}^{16} \max(0, d^2(A, P) + m - d^2(A, N_i)) \,. \tag{3.1}$$

Applying hard example mining is crucial as a large number of data points do not contain any valuable information. When applying hard negative mining only renderings of CAD models of the same category as the query image are considered. Therefore for different categories separate embedding spaces are learned using a shared encoder. Importantly, and unlike [Kuo+20], hard example mining is not applied to positive pairs. Doing so forces the network to match very different views, often not sharing any features, with each other, therefore necessarily leading to overfitting. Instead the CAD model rendering in the most similar orientation is used as an anchor. At test time we embed a masked RGB image into the embedding space and retrieve the nearest neighbour CAD model renderings which are passed on for keypoint matching and pose and shape estimation.

3.2.3 Key-Point Matching

In order to precisely estimate poses keypoint matching is performed between the masked RGB image and its retrieved CAD model rendering[Geo+19]. Finding and matching valid keypoints across the domain gap is difficult due to the different appearances of objects in RGB images. While detecting good candidate keypoints on the boundary of the CAD models is simple as they were rendered in front of a clean background, finding the corresponding keypoints in the real RGB image is more challenging due to the large variety of textures, lighting conditions and imperfect segmentation masks. We use a SuperPoint[DMR18] network for this task which was trained extensively on both detecting corners in synthetic images and

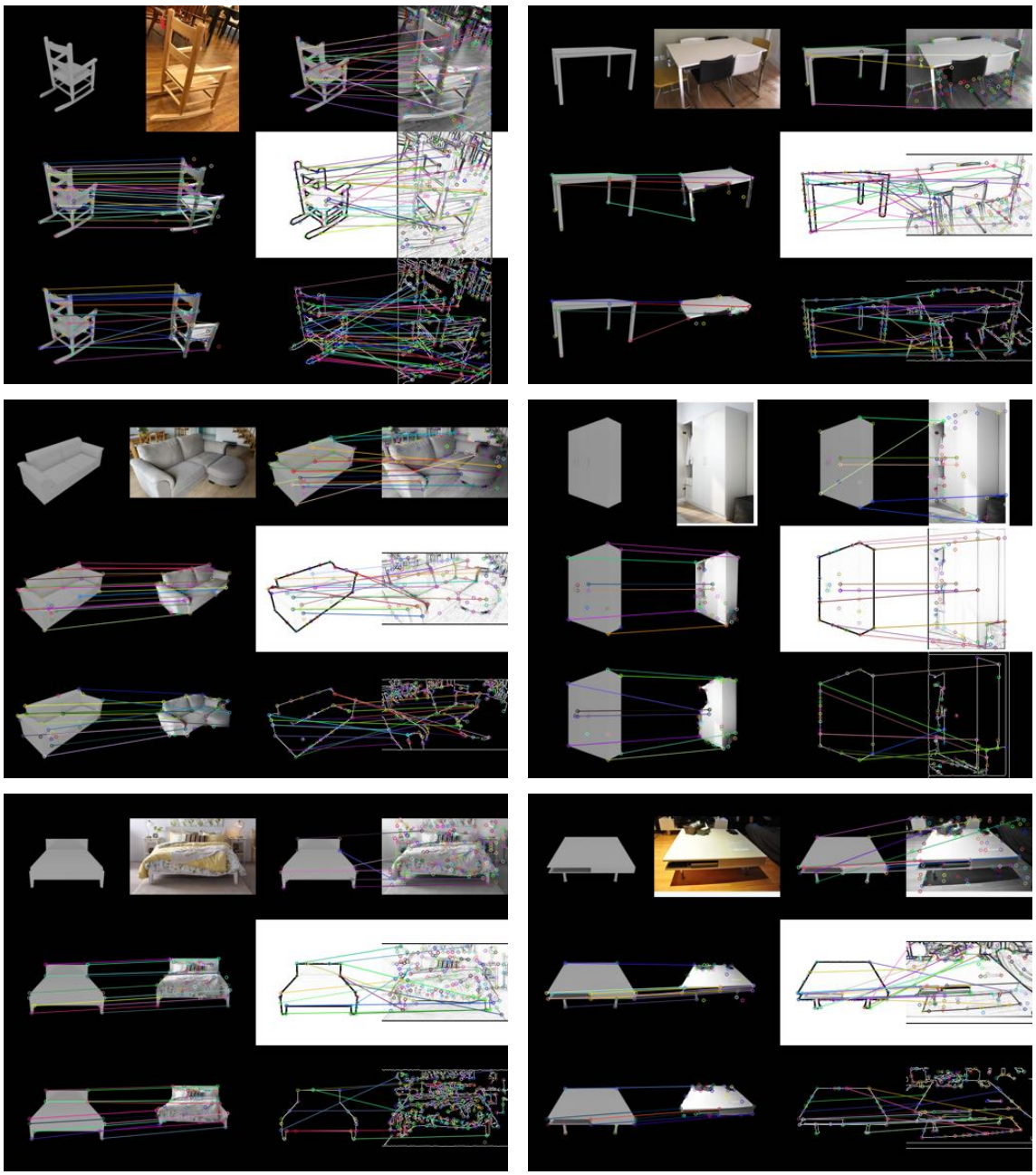


Figure 3.3: Comparison of keypoint matches obtained when applying a SuperPoint [DMR18] network to different inputs. For each example the top left images visualise the unprocessed CAD model render and the original RGB image. The images on the middle and the bottom on the left show the matches that are obtained for ground truth and predicted masks respectively. The top right shows the matches that are obtained when the input RGB image is not masked. The middle image on the right side shows matches when a pencil filter [Su+21] is applied to both the CAD model rendering and the RGB image before matching. Finally, the bottom right show the matches that are obtained when a Canny Edge detector [Can86] is applied to the images in advance.

detecting keypoints in real images, making it suitable for cross domain keypoint matching. Pre-trained network weights are used to avoid over-fitting on the train data of the comparatively smaller sized Pix3D. After detecting keypoints the corresponding feature descriptors are matched based on their L2-distance in feature space, with cross-checking in place to eliminate one-sided matches. Figure 3.3 compares matches that are obtained for different configurations including ground truth masks, predicted masks, when no masks are applied and when images are preprocessed with a pencil filter[Su+21] or Canny edge detector[Can86]. The matches for ground truth masks are generally very precise. While predicted masks still yield good matches, the number of very accurate (± 3 pixel) matches is a lot lower. This motivated us to explore other configurations which avoid introducing inaccurate object edges through predicted segmentation masks. However, one can see (right side for each example) that these do not increase the number of correct matches and often introduce a large number of false background matches. Therefore despite the inaccuracies, the proposed system uses predicted segmentation masks.

3.2.4 Pose Optimisation

The keypoint matches establish correspondences between real image pixel coordinates and 3D world coordinates in CAD model space. For the case without shape optimisation all available quadruplets of matches are sampled and their corresponding poses are computed using the UPnP-algorithm [KLS14]. The IoU overlap of the reprojected CAD models are approximated by sampling 1000 points and comparing their reprojections to the predicted object segmentation mask. Out of the sampled quadruplets the one is chosen that results in the biggest estimated silhouette overlap.

3.3 Experimental Setup

This section briefly describes the Pix3D[Sun+18] dataset that was used for training and evaluation, the AP^{mesh} metric we adopted for evaluation as well as the hyperparameters chosen.

3.3.1 Pix3D Dataset

The Pix3D[Sun+18] dataset consists of 10,069 RGB images annotated with aligned 3D CAD models (one per image). There are a total of 395 different CAD models

Figure 3.4: Data splits of the Pix3D[Sun+18] dataset first proposed by [Geo19]. Under the $S1$ split test images contain objects whose CAD models where seen during training, but which may appear under different lighting conditions, textures and generally in different scenes. For the $S2$ split test images contain object that were not seen during training.

from 9 categories (chair, sofa, table, bed, desk, bookcase, wardrobe, tool and miscellaneous). For our experiments we consider two splits originally proposed by [Geo19] (see Figure 3.4) .

S1 split

The $S1$ split randomly splits the 10,069 images into 7539 train images and 2530 test images. In this split all CAD models are seen during training and the challenge is to retrieve the correct CAD model from images containing different scenes where (possibly occluded) objects appear with new textures under varying lighting conditions.

S2 split

Under the S2 split train and test images are split such that the CAD models that have to be retrieved at test time were unseen during training. This split is more difficult as it prohibits the embedding network to simply remember CAD models and truly tests its ability to learn meaningful embeddings.

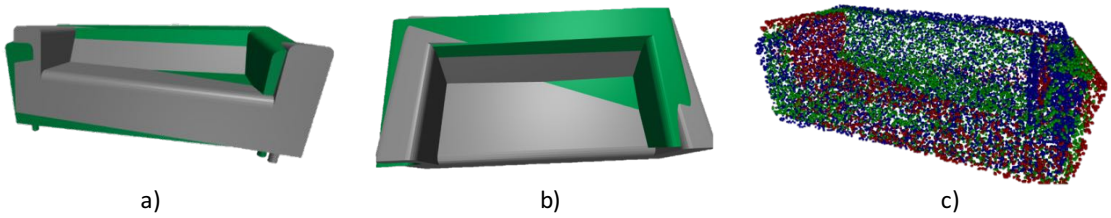


Figure 3.5: Visualisation of F1 score: a) Front view of the target shape (green) and the predicted shape (gray). b) Top view. c) Points sampled from the ground truth mesh (blue), points sampled from the predicted mesh within τ of a ground truth point (green) and points sampled from the predicted mesh not within τ of a ground truth point (red).

3.3.2 Evaluation Metric

We adopt the commonly used AP^{mesh} metric [Geo19] for evaluating the retrieved object shapes. Following the standard COCO [Lin+15] object detection protocol of AP50-AP95, we average over 10 IoU thresholds ranging from 0.50 to 0.95 in 0.05 intervals. For a given threshold the AP score is defined as the mean area under the per-category precision-recall curve where a shape prediction is considered a true-positive if its predicted category label is correct, it is not a duplicate detection, and its $F1^{\tau}$ score is greater than the IoU threshold. For a given predicted shape the $F1^{\tau}$ score is the harmonic mean of the fraction of predicted points within τ of a ground-truth point and the fraction of ground-truth points within τ of a predicted point. For our experiments we follow [Geo19; Kuo+20] in choosing $\tau = 0.3$. Similar to [Geo19; Kuo+20], for fair comparison across different object sizes we re-scale all objects such that the longest edge of the ground truth model's bounding box has length 10 before computing the $F1^{0.3}$ score.

3.3.3 Hyperparameter Settings

We render CAD models using the *Cycles* rendering engine in Blender [Com18] with 4 pointlights arranged in a square above the object and default lighting settings. CAD models are rendered at 16 regularly sampled azimuthal angles spanning 360° and 4 different elevation angles between 0° and 45°. We train the VGG encoder with a batchsize of 8 real images as each example requires 16 negative anchors and one positive anchor leading to a total of 144 images per batch. We use a learning rate of 2×10^{-6} and set the margin of the triplet-loss in Equation 3.1 to $m = 0.1$.

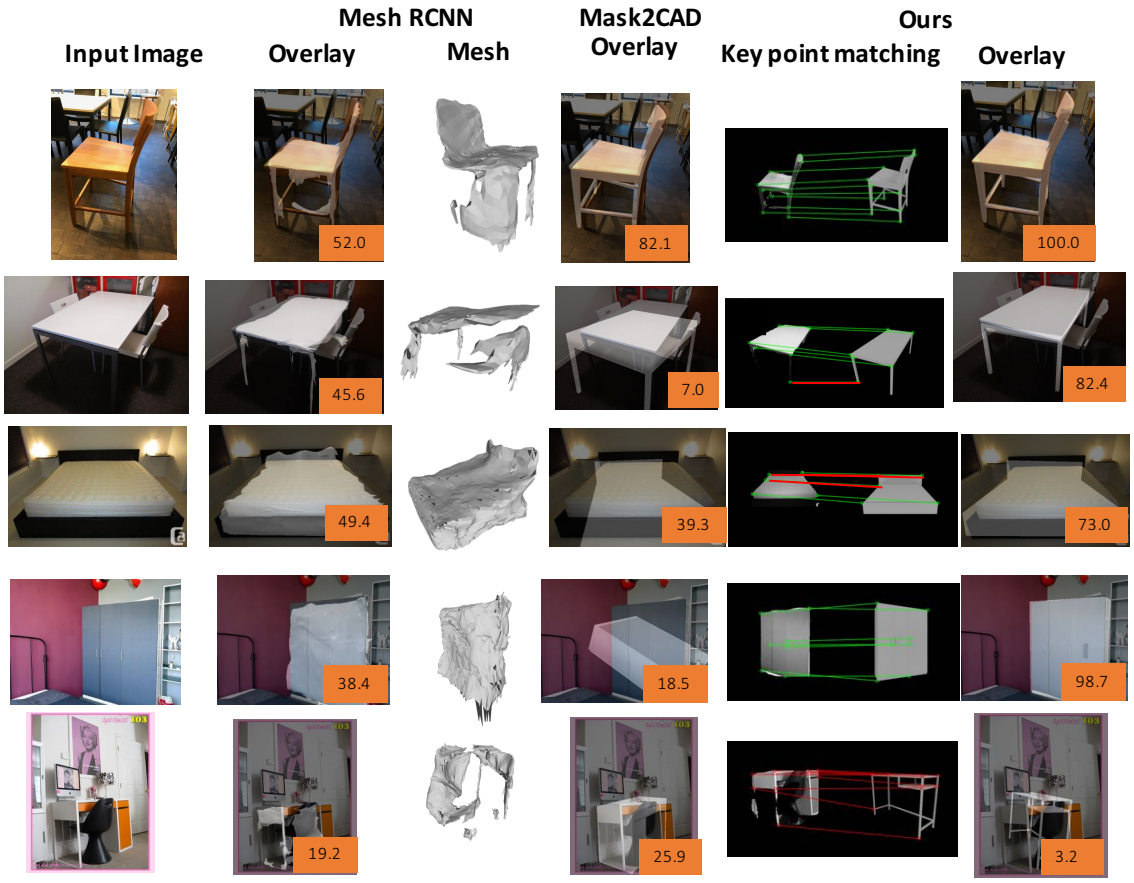


Figure 3.6: Qualitative comparison for predictions on the S2 split: The left side shows results when access to the correct CAD model is given at test time, but which were unseen at train time. The right side shows the case when no access to correct CAD models is given and retrieved CAD models have to be adapted dynamically. Numbers are F1 scores at threshold $\tau = 0.3$. In general the comparison shows that a geometric approach allows for very precise pose estimation whereas the direct prediction method of Mask2CAD [Kuo+20] is limited in the precision it can achieve. In comparison to CAD model retrieval direct mesh predictions [Geo19] are very imprecise, often failing to predict the correct topology and performing particularly poorly on the backside of objects. Row 4 shows the sensitivity of the used F1 score at threshold $\tau = 0.3$. Despite an appropriate object retrieval and very good shape adaptation imprecision in the alignments lead to a low F1 score. Finally row 5 shows a failure case of ours where poor segmentation leads to a wrong shape retrieval and correspondingly false keypoint matches resulting in a bad final pose and shape.

3.4 Experimental Results

This section showcases our experimental results. We compare against Mesh RCNN [Geo19] as well as Mask2CAD [Kuo+20]. Figure 3.6 shows the AP^{mesh} we obtain on the S1 and S2 split of Pix3D, originally proposed by Mesh R-CNN [Geo19]. Note that for a

S1	AP 50-95	bed	book-case	chair	desk	misc	sofa	table	tool	ward-robe
Mesh R-CNN	17.2	20.0	10.1	17.6	21.0	24.5	30.0	11.0	6.5	14.3
Mask2CAD	33.2	**39.4**	**42.4**	19.6	**31.6**	15.9	**55.8**	29.2	4.2	**60.3**
Ours (Mesh R-CNN) Top 1	29.2	31.7	15.7	30.6	22.5	33.6	45.5	24.7	22.2	36.4
Ours (Mesh R-CNN) Top 10	**34.4**	27.0	16.3	**34.1**	27.0	**49.2**	47.6	**33.2**	**43.7**	32.7
Ours (Swin) Top 1	31.1	21.2	19.9	29.9	25.1	35.5	41.6	25.9	43.9	36.9
Ours (Swin) Top 10	33.6	20.9	16.1	**37.2**	26.8	42.2	43.2	34.2	45.7	36.6
Ours (GT) Top 1	54.4	65.0	37.8	60.7	50.2	63.9	60.9	57.6	42.4	54.5
Ours (GT) Top 10	60.5	67.4	35.1	65.1	60.5	68.0	62.9	67.9	71.9	45.9

Table 3.1: Quantitative results on the S1 split consisting of seen objects from the Pix3D dataset. Brackets indicate the segmentation masks that were used.

fair comparison to Mask2CAD [Kuo+20] and Mesh R-CNN [Geo19] we use the ground truth z-coordinate for the final pose. While unlike [Kuo+20] our approach is not reliant on the ground truth z-coordinate it improves our performance as the very low F1 score threshold is very sensitive to displacements in the z direction arising from slight inaccuracies in the keypoint matches.

3.4.1 Seen Objects

Results on the S1 split show that particularly on seen objects CAD model retrieval is more precise than direct predictions (34.4 and 33.2 compared to 17.2 AP50-95). The proposed system outperforms Mesh R-CNN [Geo19] by a large margin in all categories. Overall we perform similar to Mask2CAD [Kuo+20]. While Mask2CAD [Kuo+20] performs well on large planar objects such as bookcases or wardrobes, we have very strong performance on high fidelity objects, such as chairs, allowing for numerous keypoint matches. When using ground truth masks compared to predicted masks we observe a large performance gain, now reaching an AP^{mesh} of more than 60 on all classes except wardrobes and bookcases. This observation is crucial as it shows the potential of our approach with improved segmentation masks. While competing approaches will also benefit from better segmentation, having accurate object silhouettes for the pose prediction is not as an integral part in their pipeline as it is for our keypoint matching and does therefore not benefit them as much.

S2	AP 50-95	bed	book-case	chair	desk	misc	sofa	table	tool	ward-robe
Mesh R-CNN	7.5	12.7	17.3	8.0	3.7	0.0	16.6	7.0	1.1	0.8
Mask2CAD	8.2	16.9	2.2	4.5	2.7	0.1	**37.8**	3.6	0.9	5.3
Ours (Mesh R-CNN) Top 1	7.4	9.6	0.5	18.5	1.6	1.5	25.7	1.6	6.8	0.5
Ours (Mesh R-CNN) Top 10	14.0	**23.4**	1.4	35.2	4.3	0.4	**39.8**	4.9	10.5	5.7
Ours (Swin) Top 1	7.0	7.3	3.2	18.7	0.6	2.1	24.3	3.6	1.6	1.2
Ours (Swin) Top 10	**15.2**	**20.6**	**4.9**	**38.5**	**4.1**	**6.9**	34.6	**7.7**	**9.6**	**10**
Ours (GT) Top 1	26.1	37.7	21.0	71.5	9.7	13.7	44.0	21.7	11.7	3.8
Ours (GT) Top 10	41.0	45.3	40.5	89.4	33.2	18.1	59.0	36.4	25.3	21.7

Table 3.2: Quantitative results on the S2 split consisting of unseen objects from the Pix3D dataset. Brackets indicate the segmentation masks that were used.

3.4.2 Unseen Objects

On the S2 split our geometric approach outperforms not only Mesh-RCNN [Geo19] but also Mask2CAD [Kuo+20] by a significant margin. While for unseen objects Mask2CAD [Kuo+20] performs very poorly for all classes except sofas and beds, we manage to maintain accurate predictions, excelling particularly at chairs where we improve over [Kuo+20] by more than 30 on the AP^{mesh} score. Note that Mask2CAD [Kuo+20] performs well on sofas, not because it is able to retrieve an unseen sofa but because for every sofa in the S2 split there is a very good fitting sofa among the seen sofas, allowing to simply retrieve that for a good performance (see supplementary material). Quantitatively the average F1 score between an unseen sofa and the best possible seen sofa is 85 which is very high compared to the average across classes which is just 64. Our results on the chair class for which on average an unseen chair has a best possible F1 score of just 63 with a seen chair show that our proposed geometric approach is able to retrieve and align unseen objects while the direct prediction method followed by Mask2CAD [Kuo+20] struggles.

3.5 Limitations

While producing accurate results the system in its current form has two drawbacks:

1. **Access to ground truth CAD models is required.** One drawback is that in order to produce accurate predictions at test time, the system requires

access to a CAD model which exactly matches the object observed in the image. While this is feasible for some scenarios in controlled environments (e.g. in factories or office spaces and houses that were previously 3D scanned), this is not realistic for more unconstrained applications. If the dimensions of the retrieved CAD models differ just slightly from those that are observed in the image, the absolute pose estimation algorithm [KLS14] will estimate wrong poses. This limitation is addressed in Chapter 4.

2. **Poses are estimated based on the silhouette overlap.** The second drawback of the proposed system is that the final pose is selected from candidate poses based on the silhouette overlap of the reprojected object with the predicted segmentation mask. We found this step to be necessary as selecting poses based on the distance of reprojected keypoints did not work as when only four[1] matches were considered many wrong poses had very small reprojected distances. The issues with selecting poses based on the silhouette overlap are threefold. First of all the process is slow as for each pose a large number of points (ca. 500 to 1000) have to be reprojected and compared to points sampled inside the segmentation mask. Secondly, the predicted segmentation masks are never perfect so that selected poses may minimise the silhouette overlap with the predicted mask, but may still be a slightly inaccurate pose (e.g because of imprecise mask edges). Finally, while the proposed keypoint-based pose estimation is in theory perfectly suited for estimating poses of partially occluded objects, using a silhouette overlap in its current form prevents this ability. Challenges related to the silhouette overlap will be addressed in future work (see Chapter 5) e.g. by using a probabilistic matching formulation which assigns soft matches that are all used jointly for predicting a distribution over object poses.

3.6 Summary

This chapter introduced a system which performs shape estimation by retrieving the best-fitting CAD model from a database. Unlike existing work we estimate the pose of the retrieved object by matching keypoints between the input image and the retrieved CAD model rendering and solving the resulting absolute pose estimation

[1] For many examples SuperPoint [DMR18] was only able to produce a maximum number of four correct matches therefore preventing us from sampling groups of five or more matches from which poses are estimated.

problem. We have demonstrated that this produces more accurate object poses compared to directly predicting them. Nevertheless, the proposed system suffers from limitations, one of which is addressed in the next chapter.

Chapter 4

Shape Estimation via Adaptation of Retrieved Objects

For realistic settings a given object will not have a perfect match among the available CAD models. This means that in order to predict accurate shapes a retrieved CAD model has to be adapted. Related work is briefly discussed in Section 4.1. Section 4.2.1 introduces a novel plane stretching formulation. This formulation is used to jointly optimise over shape and pose (see Section 4.2.2). We investigate the proposed system on a range of experiments in Section 4.3 and observe significant improvements compared to the non-stretched version of our system in Chapter 3. Just as the previous chapters this chapter too finishes with a brief summary in Section 4.4.

4.1 Related Work

Previous work most relevant to ours is [Uy+21]. [Uy+21] learn shape retrieval and deformation jointly. They optimise a retrieval and deformation module in a series of alternating steps where one is kept fixed while the other one is optimised.

- **Retrieval module.** When optimising the retrieval module an input image is encoded with a ResNet [He+16] encoder. The database of objects are represented as regions in latent space with a certain encoding. These encodings and the embedding network are optimised in an auto-decoder fashion [Par+19] such that under the current deformation predictions the deformed shape is most similar to the target shape (under the Chamfer-distance [Bar+77]).
- **Deformation module.** When training the deformation module, the retrieval

module is kept fixed and the top 10 models CAD models are retrieved. These are encoded as a whole and by their individual parts using PointNet [Cha+17b]. Together with the embedding of the target [Uy+21] predict per-part scaling and translation parameters. These part level deformations are then projected onto the space of connectivity preserving deformations and applied to obtain the deformed shape.

Interesting about this approach is that they perform deformation-aware retrieval. This means that an object is not retrieved if it is currently the best fit but if it is the best fit after an anticipated deformations. Likewise the retrieval-aware deformation allows the deformation network to focus on learning realistic deformations. By treating objects as the sum of individual parts that obey connectivity constraints [Uy+21] can perform fine-grained shape adaptations. However, currently [Uy+21] do not evaluate their system on objects in real scenes such that we can not compare our approaches directly.

4.2 Approach

The first three steps of the proposed system consisting of (i) object detection and instance segmentation, (ii) CAD model retrieval and (iii) keypoint matching are identical to the previous system described in Section 3.2. However, instead of (iv) pose optimisation we perform an optimisation over both shape and pose which is outlined below.

4.2.1 Plane Stretching Formulation

When devising a shape adaptation formulation we desire this adaptation to produce meaningful CAD model adaptions for a wide range of different shapes while maintaining a low parametric structure. On one extreme of the deformation spectrum lay global scaling operations which scale a given CAD model along 3 principal axis. While requiring only a very low number of parameters (only 3) a global scaling operation is limited in the shape adaptation it can achieve, e.g. it can never adjust proportions within a single dimension of an object such as the height of the sitting area of a chair. On the other side of the extreme are free-form deformations which allow individual per-vertex displacements. While being extremely versatile in the deformations that can be achieved, free form deformations require a great number of parameters and can often generate unrealistic shape adaptations. To alleviate some

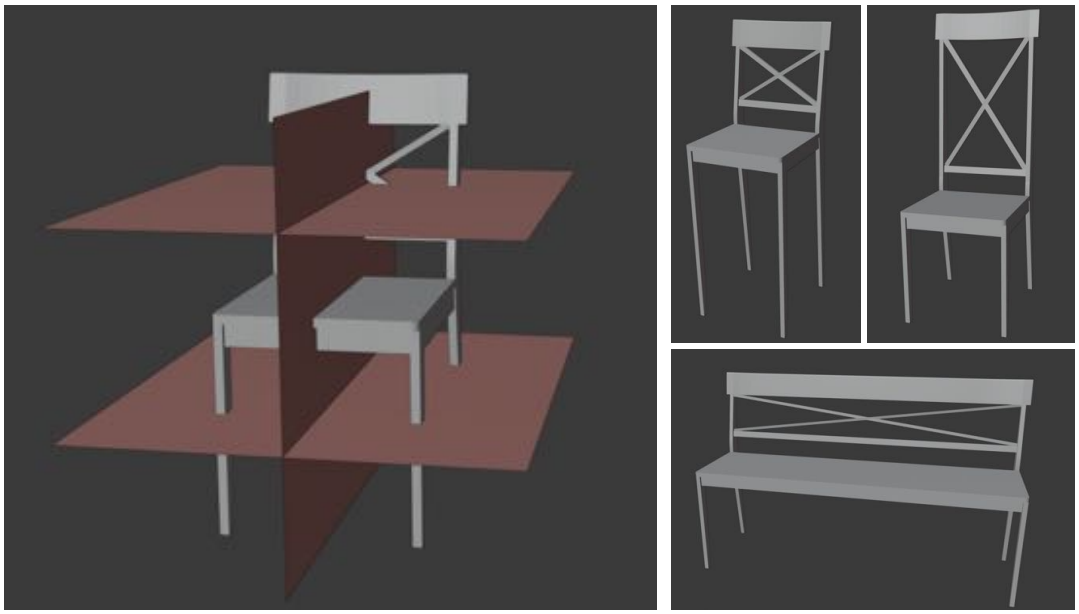


Figure 4.1: Visualisation of the proposed plane stretching formulation. Left: object with three different stretch planes. Right: deformed object after it was stretched along each of the stretch planes.

of these drawback [Yif+20] introduced a cage-deformation where the displacement of all vertices is determined from a set of control points. While being more efficient than free-form deformations cage-based deformation still require a large number of parameters[1].

In comparison to existing deformations we propose a plane stretching formulation which despite being low parametric is able to significantly modify object shapes. In this formulation an object is stretched by τ along the normal $\mathbf{n}$ of a plane P defined by $\mathbf{n} \cdot \mathbf{x} = d$. The stretched world coordinates become

$$\mathbf{x}_{\text{stretch}} = \mathbf{x} + s \times \frac{\tau}{2} \times \mathbf{n} \qquad \text{where } s = \begin{cases} 1, & \text{if } \mathbf{x} \cdot \mathbf{n} \geq d \\ 0, & \text{if } \mathbf{x} \cdot \mathbf{n} = d \\ -1, & \text{if } \mathbf{x} \cdot \mathbf{n} \leq d \end{cases}. \tag{4.1}$$

Intuitively a plane splits the vertices of a CAD into three disjoint sets: those laying on one side of the plane, those laying on the plane and those laying on the other

[1] By default the implementation by [Yif+20] uses cages consisting of 42 vertices. Therefore in order to predict the deformation from a source mesh to a target mesh 84 vertex positions or displacements have to be predicted resulting in $84 \times 3 = 252$ free parameters.

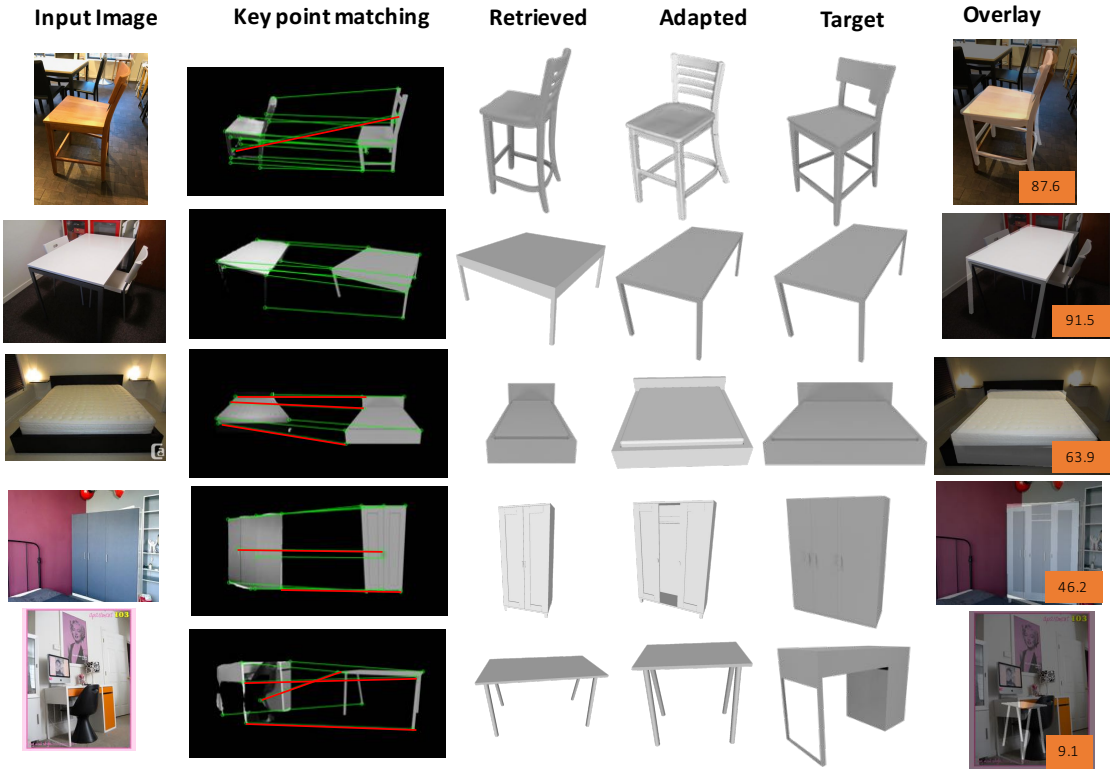


Figure 4.2: Qualitative results of stretching approach.

side of the plane. Depending on which set a given point belongs it is then displaced by τ along the normal vector of the plane, remains unchanged or is displaced by $-\tau$ along the normal vector of the plane. Figure 4.1 visualises the adapted shapes when stretching along three different planes. Note how the plane stretching allows for modifying proportions within a single dimension which is not possible with global scaling operations.

Several stretches along different planes can be repeated in succession to achieve a final deformation. Here each plane stretching requires 5 free parameters to be estimated, 3 defining the normal of the plane $\mathbf{n}$, one defining the distance of the plane from the origin d and one defining the magnitude of the stretch τ. In assuming that objects are only stretched along their principal directions, thereby specifying $\mathbf{n}$, this number can be further reduced to 2. Currently objects are only stretched along planes passing through the origin which are perpendicular to the principal axis. However, in the future one can learn to predict the orientation and position of stretch planes in the canonical object frame. This allows for shape-specific deformations e.g. increasing the width of the armrest in an armchair.

4.2.2 Joint Shape and Pose Optimisation

Previously the last step of the system consisted in estimating the pose of the retrieved CAD model from 2D-3D correspondences. Numerous solvers exist for solving this standard absolute pose estimation problem [KSS11; KLS14; LMF08]. When we additionally also perform shape adaptation, standard solvers of the absolute pose problem can not be used anymore and we instead derive and minimise a modified objective function f which minimises the reprojection error of stretched CAD model world coordinates and their corresponding pixel matches. We derive it by noting that for stretched world coordinates $\mathbf{x}_{\text{stretch}}$ one obtains reprojected pixel $\mathbf{v} \in \mathcal{R}^2$ under the perspective camera model

$$(s\,v_x, s\,v_y, s) = \mathbf{K}_{3x3}\,[\mathbf{R}_{3x3}|\mathbf{T}_{3x1}]\,[\mathbf{x}_{\text{stretch}}, 1]^T \tag{4.2}$$

for camera calibration matrix $\mathbf{K}$, rotation matrix $\mathbf{R}$ and translation vector $\mathbf{T}$. The rotation matrix is parameterised in terms of Euler angles $\theta \in \mathcal{R}^3$. For known camera intrinsics the objective function to be minimised is therefore

$$f = \sum_{j=1}^{N_{\text{matches}}} (\mathbf{u}_j - \mathbf{v}_j)^2 \tag{4.3}$$

with respect to $(\theta, \mathbf{T}, \tau)$. We use a L-BFGS[LN89] minimiser for this minimisation and initialise it with the pose of the retrieved CAD model and no stretching $\tau = 0$. For each object we perform plane stretching along three planes aligned with the three principal axis of the object and passing through the origin. Similar as before we sample groups of keypoint matches in order to reduce the sensitivity to false matches. We sample groups of 6 matches as opposed to 4 matches for the case without shape adaption as the shape adaptation introduces further free parameters. Analogously to the previous case we perform the optimisation for all sampled groups and return the result with the highest estimated silhouette overlap of reprojected object and predicted segmentation mask.

4.3 Experimental Results

We evaluate our adaptation approach on three different settings: *stretched S1 models*, *stretched S2 models* and predicting *S2 test models with S2 train models* (see Table 4.1). For the first two experiments the databases of S1 and S2 models are

modified by applying random stretching in the x, y and z direction with planes passing through the center of the object. Stretch factors are sampled from a uniform distribution on the interval $[-0.2, 0.3]$ and multiplied by the respective 3D-bounding box side length of the object. These stretches significantly alter the shapes of the CAD models and therefore require successful shape adaptation for precise predictions. The third experiment investigates the systems capability of adapting CAD models to match entirely different ones. Here the shape of S2 test CAD models has to be estimated using the disjoint set of S2 train CAD models. This requires the robust retrieval of similar, but different CAD models as well as keypoint matching that identifies part correspondences, but which may have local, visual differences.

4.3.1 Stretched S1 Models

The results for estimating shapes from the S1 test split when only access to randomly stretched CAD models is provided can be found in the first two rows of Table 4.1. We note that the AP score we achieve without stretching is very low. This highlights that the random shape stretches we applied to the CAD models in the database were substantial enough to require stretching at test time for accurate predictions. When performing stretching we observe significant improvements in all object categories.

4.3.2 Stretched S2 Models

We repeat the same experiment as above with the S2 split under which the CAD models at train and test time are split into disjoint sets. Compared to the S1 split we here observe a much smaller improvement on the average AP score. While for the stretched S2 models stretching improves the shape predictions on most classes (e.g. bookcases, chairs, tables) it fails on some categories, most notably on sofas. As was shown above for sofas very accurate shape models already exist in the non-stretched part of the database, such that in this case allowing stretching from sometimes poor segmentation masks and corresponding keypoint matches can deteriorate the shape. In general the main issue when moving from the S1 split to the S2 split is the worsening of the segmentation mask as segmentation networks have difficulties in accurately segmenting unseen objects. Worse segmentation masks subsequently lead to slightly worse shape retrieval, but more importantly to inaccurate keypoint detection and matches. As the pose and shape optimisation is very sensitive to pixel misalignments these significantly reduce the quality of the estimated shapes.
We investigating the performance of the system on the S2 split when ground truth

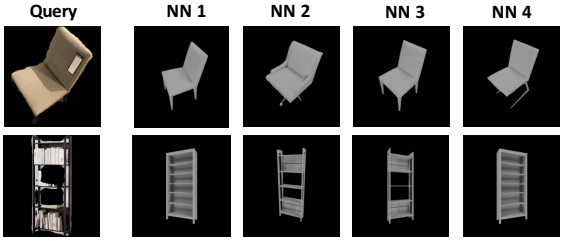
		AP50-95	bed	book case	chair	desk	misc	sofa	table	tool	wardrobe
S1 stretched	Ours (Swin) no stretching	4.4	8.7	3.1	4.5	2.6	1.0	13.8	3.3	0.7	1.9
	Ours (Swin) with stretching	15.5	16.8	12.7	16.4	15.4	8.5	29.3	17.6	5.1	17.4
S2 stretched	Ours (Swin) no stretching	8.5	22.8	0.9	5.8	0.2	0.0	34.7	3.5	2.0	6.4
	Ours (Swin) with stretching	9.3	21.0	7.7	17.6	2.3	1.1	24.1	5.0	0.5	4.4
S1 models to S2	Mask2CAD	**6.5**	14.0	**2.2**	3.2	0.2	0.0	**35.4**	1.2	0.6	1.6
	Ours (Swin) no stretching	6.4	13.4	0.2	4.7	0.1	0.0	29.8	1.0	7.9	0.0
	Ours (Swin) with stretching	**6.5**	**18.6**	2.1	**4.6**	**0.9**	0.0	25.3	**2.9**	**1.7**	**2.2**
	Ours (GT) no stretching	7.2	12.0	4.5	4.7	0.0	0.0	37.0	2.3	3.9	0.0
	Ours (GT) with stretching	11.6	24.9	11.3	4.6	3.3	4.7	35.8	11.0	2.4	6.7

Table 4.1: Quantitative results on Pix3D when no access to correct models is provided at test time. For the first two rows we randomly stretch CAD models in our database along all 3 principal direction and our method has to recover the original shape. For the last row S2 CAD models have to be estimated when the retrieval network has no access to the correct models and differing CAD models have to be adapted. Experiments on a stretched version of S1 models demonstrate that shape adaptation substantially improves the shape predictions. While on the S2 we can observe improvements for certain classes the overall accuracy gain is smaller, with the main reason being poorer segmentation quality which prevents the matching network from successfully establishing correspondences.

masks are provided. For this purpose we introduce further versions of the CAD models which are randomly stretched along one or two principal axis. Figure 4.3 b visualises the average AP mesh score we obtain as CAD models in the database are increasingly deformed. We note that without stretching at test time the AP mesh score rapidly decreases from 68 for the original models to 4 when models were stretched along all three principal axis. Comparing this to the case were stretching is allowed at test time we can retain a high AP mesh score of 48 even when objects were previously stretched along all three principal axis.

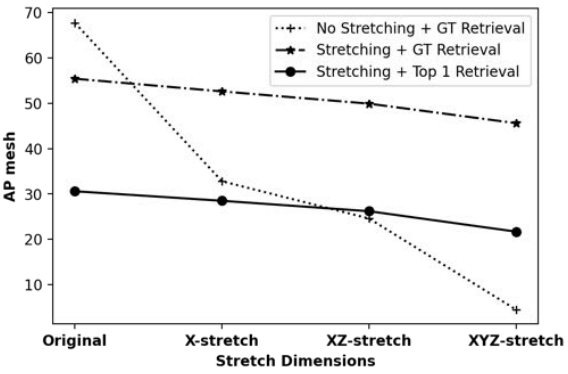
4.3.3 Estimating S2 Test Models from S2 Train Models

Finally, we investigate how well S2 test models can be approximated when only access to the disjoint set of S2 train models is provided. Qualitative prediction of the model using stretching can be seen in Figure 4.2. The first three rows show examples were the proposed stretching successfully adapts retrieved shapes to match the target shapes leading to a high F1 score. While the adapted shape in the fourth row is still very similar to the target shape small pose misalignment lead to a reduced F1 score. Finally, the last row shows an example of poor segmentation leading to a



	Predicted Masks		GT Masks	
	Top 1	Top 10	Top 1	Top 10
S1	65 %	85 %	87 %	98 %
S1 stretch XYZ	44 %	69 %	70 %	90%
S2	24 %	52 %	62 %	91 %
S2 stretch XYZ	18 %	48 %	49 %	80 %

a)



b)

Figure 4.3: a) Retrieval accuracy for selected CAD model splits. When considering the top 10 nearest neighbours the retrieval network is able to return completely unseen CAD models in over 50% of cases. Note that different renderings of the same CAD model are considered as different nearest neighbours. b) Ablation experiments on the proposed object stretching with ground truth masks. We plot the average AP mesh score as a function of increasing shape deformations of S2 CAD models. On the left no deformations were performed while on the right objects were stretched along the x,y and z direction. With increasing deformation simple object retrieval quickly becomes inaccurate, while the proposed stretching is able to maintain a high accuracy.

sub-optimal retrieval result with poor keypoint matches.

Investigating the results qualitatively we observe that similar to the previous experiment on the S2 split the improvement from using stretching is very small, the reason being again poor segmentation masks and the associated inaccurate correspondences. Comparing to Mask2CAD [Kuo+20] who do not perform shape adaptation we note that our approach compares favourably for all classes except bookcase and sofa. The strong performance Mask2CAD [Kuo+20] achieves on sofas is again due to the high similarity of S2 train and test sofas (see Appendix A for more details). When using ground truth masks instead of predicted masks we notice large improvements from allowing stretching (e.g bed from 12.0 to 24.9, bookcase from 4.5 to 11.3 or table from 2.3 to 11.0). Figure 4.4 provides some qualitative examples where shape predictions fail when using predicted mask but are accurate for ground truth masks.

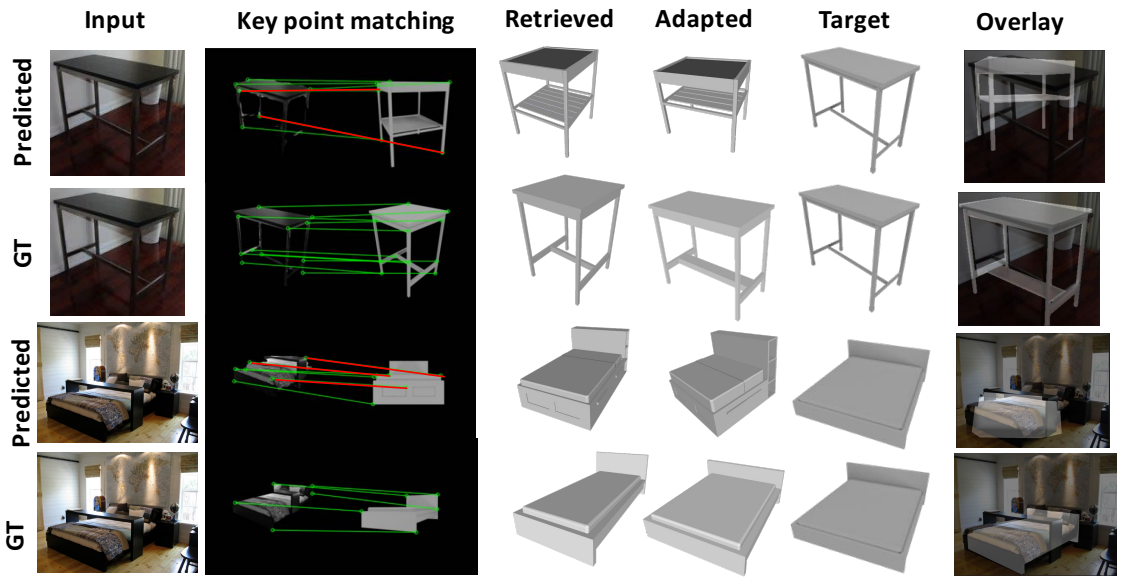


Figure 4.4: Visual comparison of shape prediction with predicted (row one and three) and ground truth (row two and four) segmentation masks.

4.4 Summary

This chapter introduced a novel plane stretching approach. It was motivated from the need for low parametric shape adaptations that allow for realistic object modifications. We have demonstrated the usefulness of this formulation on a range of experiments. However, we note that the shape and pose optimisation we perform is sensitive to inaccurate correspondences arising from poor segmentation masks. In future work we hope to increase the robustness of our system by using denser correspondences and by improving indoor object segmentation.

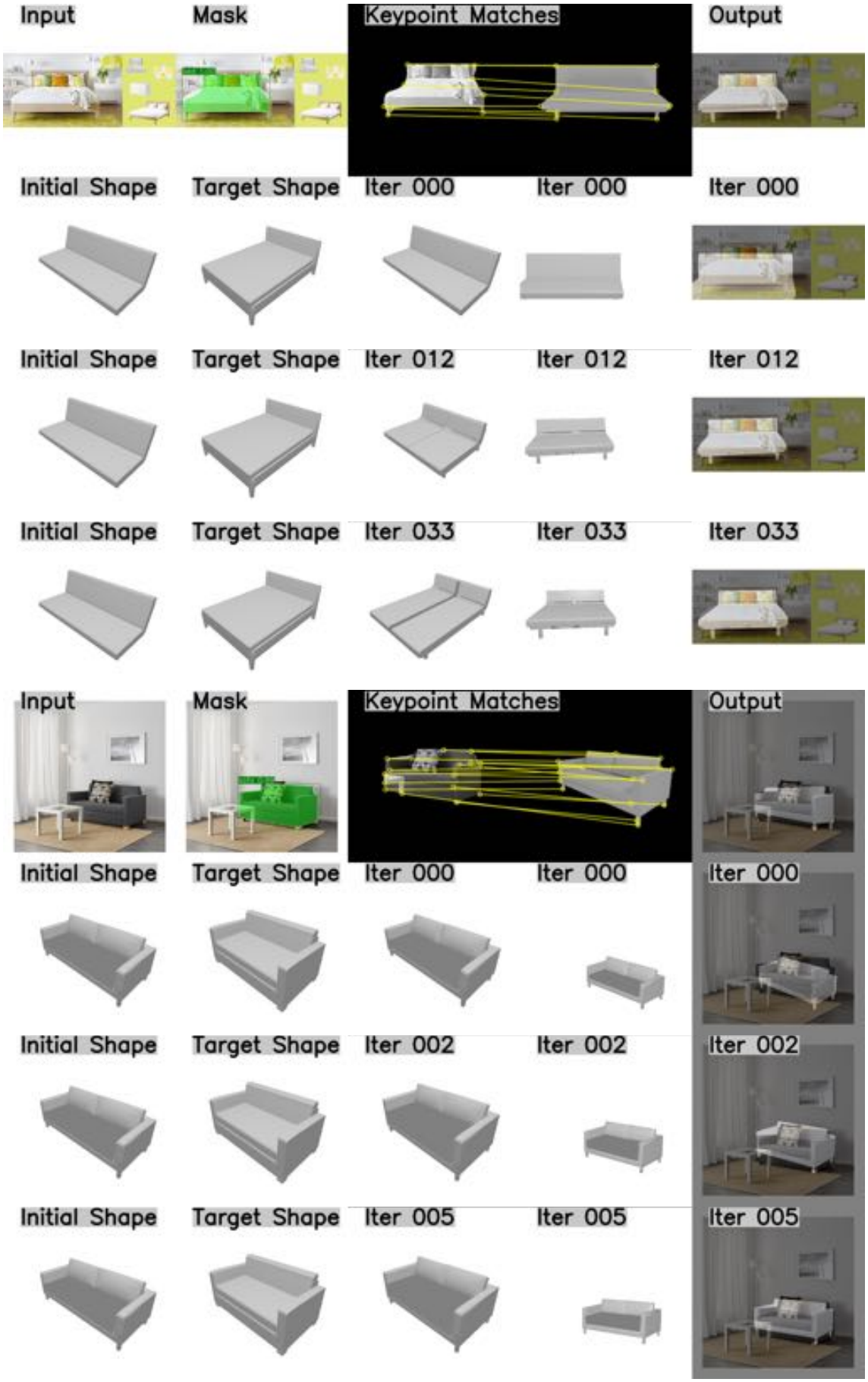


Figure 4.5: Visualisation of shape deformation at different iterations when using predicted masks.

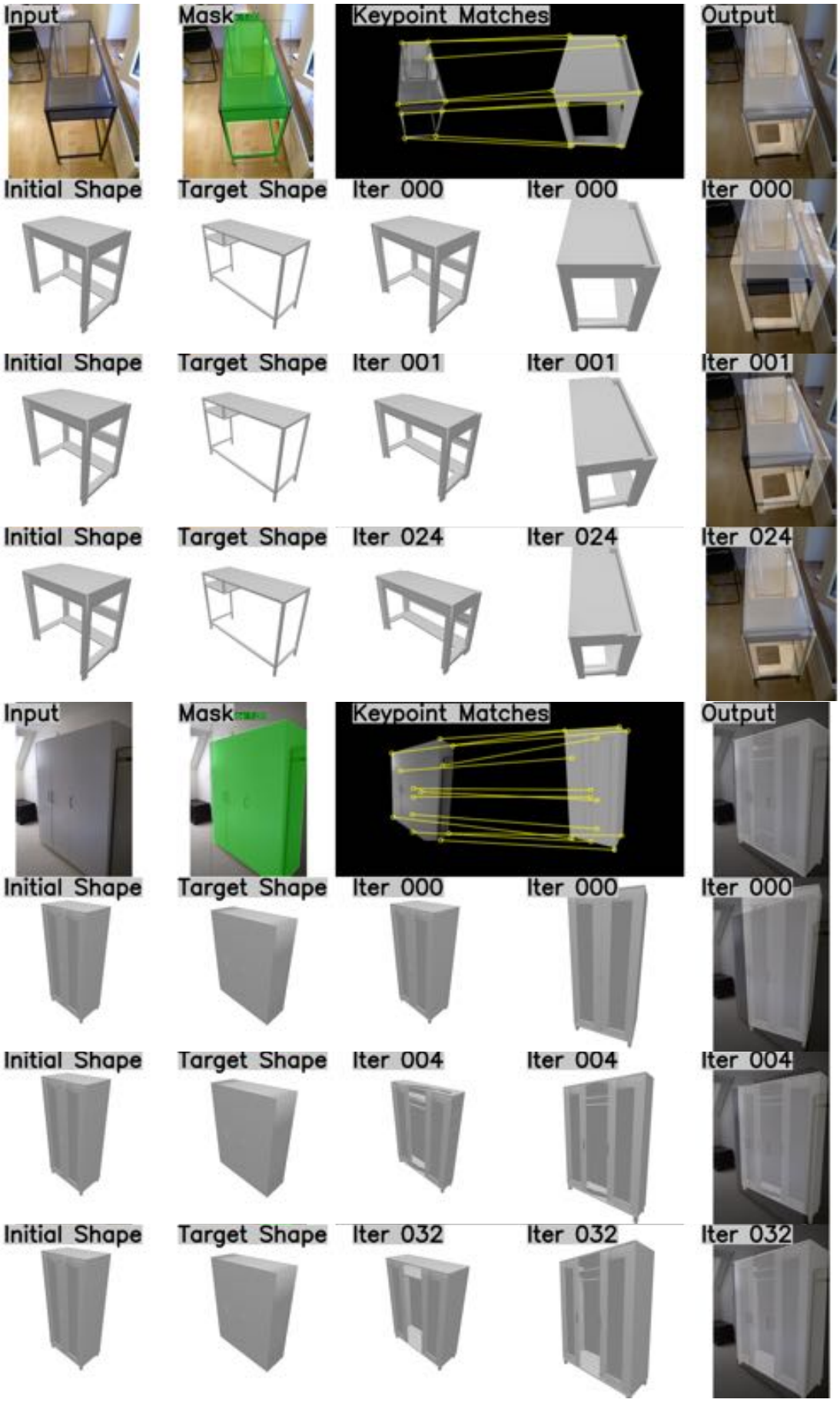


Figure 4.6: Visualisation of shape deformation at different iterations when using ground truth masks.

Chapter 5

Discussions and Future Work

This chapter concludes this report. The first Section 5.1 presents a discussion of the topics and ideas that were presented. The second Section 5.2 outlines possible future work and an associated timeline.

5.1 Summary

This report addressed the problem of 3D shape estimation of static objects from a single RGB images. To date a wide variety of approaches aim to tackle this problem by directly predicting a 3D shape [Cho+16b; FSG16; ZKG20; Wan+18; Geo19; Pan+19; Nie+20; Den+20]. This report argues that a more reliable and accurate approach is based on retrieving an existing CAD model from a database. Particularly in man-made environments governed by high regularities a small set of CAD models can approximate a large number of observed shapes. Current retrieval approaches [Kuo+20; Eng+21] rely on directly predicting object poses. We demonstrate that more accurate predictions can be obtained through a geometric approach for computing object poses. For this we establish correspondences between the input image and the retrieved CAD model render. These correspondences are then used in turn to compute the object pose analytically. In transforming the pose estimation problem to a keypoint matching problem we simplify the learning task for the network. Whereas previously the network had to regress the pose of an object directly from an image (which is difficult and imprecise) the network can now focus its predictive power on the simpler task of pattern matching task which traditionally have been the strengths of neural networks.

While many objects can be approximated using just a small database of CAD models precise predictions often require some form of shape adaptation. Existing work

on shape deformation [Yif+20; Uy+21; Jac+18] often require estimating a large number of free-parameters. In order not to rely on large number of parameter predictions from a network but rather geometric correspondences we introduce a low parametric plane stretching formulation. While only requiring few parameters it is nevertheless capable of producing significant and realistic shape adaptations. We show the usefulness of the proposed stretching procedure in a range of experiments. While the work detailed in this report can be improved in numerous ways (as outlined in Section 5.2) we believe that in general adapting retrieved CAD models is a promising avenue for future research as it combines the reliability of object retrieval with the expressiveness of generative approaches.

5.2 Future Work

There exist numerous ways in which the displayed work can be extended and improved. Section 5.2.1 provides brief overviews over selected directions for future research. Section 5.2.2 sketches a timeline for researching some of the proposed ideas.

5.2.1 Avenues for Improving Geometric Shape Estimation

The ideas proposed here are presented in estimated order of importance for improving 3D shape estimation.

Dense Matching

One of the key weaknesses of the current system are the low number of correct correspondences that are established between the retrieved CAD model renders and the masked input image. On the S2 split of the Pix3D[Sun+18] dataset given perfect retrieval 26% of examples have less than 3 correct keypoint matches making an accurate pose computation impossible. For more realistic retrieval scenarios this number is even higher. Such few correspondences make the system sensitive to false matches and limit the capabilities of any deformation procedure. We therefore propose to perform denser matching procedures. Recent work [Gra+20] learns differentiable rendering for 3D pose refinement. Comparing real RGB images and rendered images of CAD models they establish dense correspondences in feature space and use these to directly inform a gradient-based optimisation of the 3D pose. A similar technique can be adapted to our problem where instead of optimising over

Figure 5.1: Given an initial pose estimate a) [Gra+20] establish dense correspondences in feature space b) and use these to inform their pose updates to obtain a final pose c). Figure from [Gra+20].

the 3D pose we optimise jointly over shape and pose. However, [Gra+20] restrict their correspondence predictions to "visible object regions with significant geometric discontinuities". This effectively limits their correspondence predictions to edges. While we understand that this step was done to increase the accuracy of the predicted correspondences, we also plan to match entire object surfaces (similar to Blob detection and matching). Surface normal predictions may be useful for this task as they are stable over single object surfaces.

Probabilistic Matching

Another drawback of the current implementation is that correspondences are established through hard assignments. Matches are established between keypoints if they are closest to each other in terms of the $L2$-distance. However, in this manner the uncertainty of a match (does this match establish correspondence between the same object region?) as well as its precision (how accurate is the established correspondence?) are not taken into account. The current implementation deals with incorrect and imprecise matches by sampling groups of matches and scoring the computed pose based on the estimated silhouette overlap. However, directly incorporating the uncertainty of estimated matches is necessary for better informed pose estimates (e.g. like [Vak+21]).

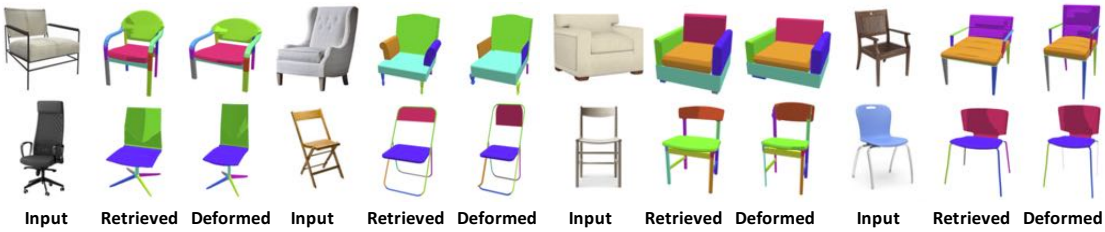


Figure 5.2: Visualisation of the results obtained with the part-aware deformation procedure by [Uy+21]. Figure from [Uy+21].

Increased CAD Model Diversity

Another possible extension of the presented work is to extend the range of object shapes that can be estimated. One way of approaching this problem is to increase the size of the CAD model database on which the system is trained and to which it has access at test time. Again and again the history of deep learning has shown that large improvements can be obtained by simply increasing the amount of training data. For the problem at hand this can be achieved most easily through the usage of synthetic datasets as explained in Section 2.3.2. A second approach to enable a larger CAD model diversity is to use generative models [Wu+19; Mo+19b]. Those can be sampled in advance to produce large databases capturing a wide range of different shapes. A third way to enable the system to approximate more diverse shapes is to introduce a more powerful shape adaptation procedure. While this is difficult to achieve with the current sparse matching procedure, denser matching as outlined in Section 5.2.1 may pave the way for more versatile deformations. Such deformations may be part-based as demonstrated by [Uy+21] which allow for extremely fine-grained shape adaptations while still leading to overall valid object shapes.

Modelling Object Dependencies

Better object shape predictions may also be achieved by estimating them jointly for all objects detected rather than separately. Intuitively, such an approach may learn correlations appearing abundantly in realistic room scenes. These include intra-class relations (For example chairs surrounding a dining table are most likely of the same kind. Particularly for estimating the shapes of occluded chairs learning object relations may help a lot.), inter-class relations (For example recognising a desk may influence the retrieval process of the most similar chair and its estimated pose) and object-room layout relations (A network may learn that most objects are aligned with the principal axis of a room, particularly objects close to walls.). Existing work

modelling such object dependencies, albeit in slightly different form, are for example [Nie+20; Ave+19].

Improved Indoor Segmentation

An obvious, yet important way to improve object shape estimation from a single RGB image is improved indoor segmentation. An accurate segmentation mask is crucial for successful CAD model retrieval, but even more so for successful keypoint matching. The segmentation networks trained as part of the current system often produce segmentation outputs which are blurry and imprecise around object edges. However, these segmentation mask edges are of particular importance for any matching procedure, therefore improving them would be very beneficial. In general the quality of indoor segmentation seems to lag behind that of outdoor segmentation. A likely reason for this are the more diverse lighting, textures and object variability found in indoor scenes. As demonstrated by [McC+17] improvements on object segmentation may therefore be obtained through extensive training on synthetic data. A more specific proposal for improving object segmentation for our purposes makes use of existing work on line predictions [Gu+21]. We note that for many RGB images line predictions of the pre-trained model [Gu+21] are more precise along object edges than our segmentation networks leading to interesting opportunities of combining these (see Figure 5.3).

Temporal Frame Combinations

This extension goes beyond the originally posed problem of shape estimation from a single RGB image and as such appears last in this list. It is relevant as many applications for example in robotics or augmented reality provide not just a single RGB image but rather a continuous stream. Such applications motivate loosening the constraint of using just a single RGB images to using multiple frames of a video. Crucially this allows resolving the scale-depth ambiguity which otherwise can only be solved by imposing other constraints (e.g. that objects are grounded on the floor). Furthermore, combining multi-view predictions greatly improves the accuracy of predicted object poses (as demonstrated by [Man+20]) and may allow for more precise shape adaptations.

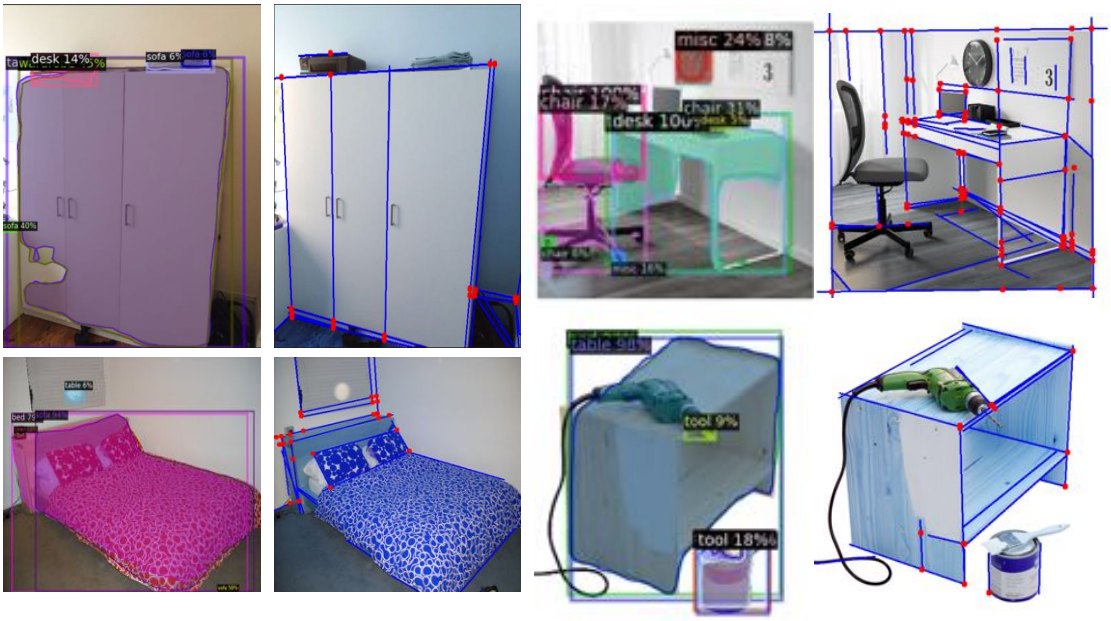


Figure 5.3: Side-by-side view of instance masks predicted by Mesh-RCNN [Geo19] and line segments predicted by [Gu+21] (without retraining) on images from the S2 split of Pix3D [Sun+18]. Line predictions along object edges are often more precise than the corresponding segmentation masks. In future work line predictions may therefore be used to refine the boundaries of segmentation masks.

Dates	Goal
Sep 2021 - Feb 2022	Probabilistic problem formulation
Mar 2022 - Aug 2022	Additional shape adaptations and generative models
Sep 2022 - Feb 2023	Holistic scene understanding
Mar 2023 - Aug 2023	Consolidation of results and final thesis write-up

Table 5.1: Timeline for future work

5.2.2 Timeline/Timetable

After discussing promising future work in Section 5.2.1 Table 5.1 provides an estimated timeline for researching the presented extensions. They are presented in more detail in the following.

Probabilistic problem formulation

Many object shapes and poses that are observed from a single view are ambiguous, e.g. there is an inherent scale-depth ambiguity present. Those ambiguities become more pronounced if objects are partially occluded or viewed from such an angle that large parts of the object are facing away from the camera. For many realistic scenarios predicting a set of possible shape and pose explanations is appropriate,

rather than predicting just a single one. This naturally calls for a probabilistic problem formulation which can be broken down into four distinct tasks.

1. **Specifying a joint probability distribution.** The first task will be to specify a joint probability distribution over object pose, deformation parameters, pixel bearings, pixel depths and 3D world coordinates. This probability distribution should attain large values if a given configuration is geometrically consistent and low values if it is not. One simple way would be to specify a fully factorised distribution where each potential encodes the error of the projected pixel bearing as in [Zim21]

2. **Incorporating dense matches.** By specifying a joint probability distribution we can model the uncertainty of proposed matches. This allows us to relax the need of perfect matches so that matches are not limited to corners (as they currently are), but can lie on object edges or surfaces. This allows for a large number of dense matches.

3. **Devising a method for performing inference and learning over the probability distribution.** Crucially we need to be able to perform inference over the proposed joint probability distribution to obtain a distribution over just the pose and stretch parameters. [Zim21] show that when only object poses are considered inference can be performed in a brute force way by densely sampling object rotations.

4. **Speeding up the inference.** While as a first step any method that allows us to perform inference is useful, we eventually need to devise a method that allows for fast inference. Variational Bayesian methods may be used for this purpose.

Additional shape adaptations and generative models

While the proposed plane stretching formulation is effective, the number of shapes that can be approximated with it is still limited. In order to better estimate a wide range of different shapes we will explore using other adaptation procedures as well as generative models.

1. **Different shape adaptations.** One limitation of the current plane stretching formulation is that it does not work well for curved objects. One possible deformation procedure that allows for more realistic shape adaptations of curved

objects are cages. [Jak+20] show how neural cages[Yif+20] can be controlled from a small set (less than 10) of 3D control points on the object. [Jak+20] learn suitable control points in an unsupervised manner, therefore requiring no additional annotations of CAD models. This means that the system developed by [Jak+20] can be directly incorporated into our framework. Rather than optimising over plane stretching parameters, we can optimise over the position of 3D control keypoints to adapt shapes.

2. **Increase database size using generative models.** An alternative approach for better estimating a wide range of different shapes is to increase the size of the CAD model database, so that every shape observed has an almost perfectly matching shape in the database. Expanding a database to such an extent manually is extremely expensive and not feasible. We will therefore explore using generative models for shapes[Wu+19; Mo+19b] for this task. Using a sampling strategy which produces a very, diverse but still realistic set of shapes will be crucial.

3. **Sampling from generative models directly.** The problem with increasing the size of the database is that retrieving shapes will be slower. In order to avoid brute force comparisons between query and all database embeddings, one can devise smart sampling strategies. These may be based on the learning-to-learn[And+16] idea where networks are used to predict gradients. One strategy would be to train a network that given a query embedding and a retrieved shape embedding estimates updates for the retrieved shape embedding directly. Performing shape embedding updates multiple times allows to quickly retrieve suitable shapes without exhaustively comparing against all shapes in a database.

Holistic scene understanding

Our current system makes object shape and pose predictions independently of each other. When considered jointly these predictions can be unrealistic e.g. as objects are not aligned or intersecting each other. In general we anticipate that jointly predicting object poses will be more crucial than joint shape predictions. This is because poses are highly correlated e.g. objects are aligned with principal room axis, whereas shapes depend more on the type of room (e.g. whether a chair is in a kitchen or office) than other shapes in the room. Building a holistic model can be achieved at three different levels:

1. **Extracting pairwise relations.** We can use datasets such as ScanNet [Dai+17] or Matterport 3D [Cha+17a] to collect pairwise relations of relative poses and support relationships. These can then be used as priors to the probabilistic model e.g. it is likely that a table is parallel to a wall and a monitor is supported by a table.

2. **Modelling a scene-graph.** A more sophisticated solution could be to model object-object and object-scene (e.g. the wall or floor) relations using a graph [Ave+19]. While in theory these relations can be learned directly, modelling them explicitly will reduce the amount of training data that is needed to learn them. Initially nodes in the graph may contain the object poses as predicted from individual object predictions. By performing a sequence of graph convolutions individual object nodes can accumulate information about the poses of surrounding objects and learn to update their poses to produce a globally more consistent room layout (see [Kan+18] for an example of an iterative pose regression module).

3. **Learning a generative model for entire rooms.** The ultimate goal for holistic scene understanding is to devise a generative model not for individual shapes, but for an entire room. Currently learning accurate generative models of entire scenes is extremely difficult due to the large variability of rooms and the existing objects within them. However, in the future larger datasets and more advanced learning techniques may make learning such generative models possible.

Bibliography

[Bar+77] H. G. Barrow et al. "Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching". In: *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI'77. Cambridge, USA: Morgan Kaufmann Publishers Inc., 1977, pp. 659–663.

[Can86] John Canny. "A Computational Approach to Edge Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (1986), pp. 679–698. DOI: 10.1109/TPAMI.1986.4767851.

[LN89] Dong C. Liu and Jorge Nocedal. "On the Limited Memory BFGS Method for Large Scale Optimization". In: *MATHEMATICAL PROGRAMMING* 45 (1989), pp. 503–528.

[Pow94] M. J. D. Powell. "A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation". In: *Advances in Optimization and Numerical Analysis*. 1994, pp. 51–67. DOI: 10.1007/978-94-015-8330-5_4. URL: https://app.dimensions.ai/details/publication/pub.1046127469.

[Che+03] Ding-Yun Chen et al. "On Visual Similarity Based 3D Model Retrieval". In: *Computer Graphics Forum* 22.3 (2003), pp. 223–232. DOI: https://doi.org/10.1111/1467-8659.00669. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-8659.00669.

[Gao+03] Xiao-Shan Gao et al. "Complete solution classification for the perspective-three-point problem". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.8 (2003), pp. 930–943. DOI: 10.1109/TPAMI.2003.1217599.

[DT05] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[LMF08] V. Lepetit, F. Moreno-Noguer, and P. Fua. "EPnP: An Accurate O(n) Solution to the PnP Problem". In: *International Journal of Computer Vision* 81 (2008), pp. 155–166.

[MH08] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605. URL: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

[Che+10] Gal Chechik et al. "Large Scale Online Learning of Image Similarity Through Ranking". In: *J. Mach. Learn. Res.* 11 (Mar. 2010), pp. 1109–1135. ISSN: 1532-4435.

[SF10] Yann Savoye and Jean-Sébastien Franco. "CageIK: Dual-Laplacian Cage-Based Inverse Kinematics". In: *Articulated Motion and Deformable Objects, 6th International Conference, AMDO 2010, Port d'Andratx, Mallorca, Spain, July 7-9, 2010. Proceedings*. Ed. by Francisco J. Perales López and Robert B. Fisher. Vol. 6169. Lecture Notes in Computer Science. Springer, 2010, pp. 280–289. ISBN: 978-3-642-14060-0.

[KSS11] Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation". In: *CVPR 2011*. 2011, pp. 2969–2976. DOI: 10.1109/CVPR.2011.5995464.

[NF12] Pushmeet Kohli Nathan Silberman Derek Hoiem and Rob Fergus. "Indoor Segmentation and Support Inference from RGBD Images". In: *ECCV*. 2012.

[Gir+13] Ross B. Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *CoRR* abs/1311.2524 (2013). arXiv: 1311.2524. URL: http://arxiv.org/abs/1311.2524.

[LPT13] Joseph J. Lim, Hamed Pirsiavash, and Antonio Torralba. "Parsing IKEA Objects: Fine Pose Estimation". In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 2992–2999. DOI: 10.1109/ICCV.2013.372.

[Uij+13] Jasper Uijlings et al. "Selective Search for Object Recognition". In: *International Journal of Computer Vision* 104 (Sept. 2013), pp. 154–171. DOI: 10.1007/s11263-013-0620-5.

[Aub+14] Mathieu Aubry et al. "Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models". In: *CVPR*. 2014.

[KLS14] Laurent Kneip, Hongdong Li, and Yongduek Seo. "UPnP: An Optimal O(n) Solution to the Absolute Pose Problem with Universal Applicability". In: *ECCV (1)*. 2014, pp. 127–142.

[Cha+15] Angel X. Chang et al. *ShapeNet: An Information-Rich 3D Model Repository*. 2015. arXiv: 1512.03012 [cs.GR].

[Li+15] Yangyan Li et al. "Joint Embeddings of Shapes and Images via CNN Image Purification". In: *ACM Trans. Graph.* 34.6 (Oct. 2015). ISSN: 0730-0301. DOI: 10.1145/2816795.2818071.

[Lin+15] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].

[Lop+15] Matthew Loper et al. "SMPL: A Skinned Multi-Person Linear Model". In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34.6 (Oct. 2015), 248:1–248:16.

[Ren+15] Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes et al. Vol. 28. Curran Associates, Inc., 2015. URL: https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

[SZ15] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[SLX15] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. "SUN RGB-D: A RGB-D scene understanding benchmark suite". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 567–576. DOI: 10.1109/CVPR.2015.7298655.

[And+16] Marcin Andrychowicz et al. "Learning to learn by gradient descent by gradient descent". In: *NIPS*. 2016.

[Cho+16a] Christopher B Choy et al. "3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2016.

[Cho+16b] Christopher B. Choy et al. "3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction". In: *CoRR* abs/1604.00449 (2016). arXiv: 1604.00449.

[FSG16] Haoqiang Fan, Hao Su, and Leonidas Guibas. *A Point Set Generation Network for 3D Object Reconstruction from a Single Image*. 2016. arXiv: 1612.00603 [cs.CV].

[He+16] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[ISS16] Hamid Izadinia, Qi Shan, and Steven M. Seitz. "IM2CAD". In: *CoRR* abs/1608.05137 (2016). arXiv: 1608.05137.

[Red+16] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[Arm+17] I. Armeni et al. "Joint 2D-3D-Semantic Data for Indoor Scene Understanding". In: *ArXiv e-prints* (Feb. 2017). arXiv: 1702.01105 [cs.CV].

[BNG17] Bert De Brabandere, Davy Neven, and Luc Van Gool. *Semantic Instance Segmentation with a Discriminative Loss Function*. 2017. arXiv: 1708.02551 [cs.CV].

[Cha+17a] Angel Chang et al. "Matterport3D: Learning from RGB-D Data in Indoor Environments". In: *International Conference on 3D Vision (3DV)* (2017).

[Cha+17b] R. Qi Charles et al. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 77–85. DOI: 10.1109/CVPR.2017.16.

[Dai+17] Angela Dai et al. "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes". In: *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*. 2017.

[McC+17] John McCormac et al. "SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?" In: (2017).

[Vas+17] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[Zho+17] Bolei Zhou et al. "Scene Parsing through ADE20K Dataset". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5122–5130. DOI: 10.1109/CVPR.2017.544.

[Com18] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam, 2018. URL: http://www.blender.org.

[DMR18] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. "SuperPoint: Self-Supervised Interest Point Detection and Description". In: *CVPR Deep Learning for Visual SLAM Workshop*. 2018.

[Gro+18] Thibault Groueix et al. "AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation". In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[He+18] Kaiming He et al. *Mask R-CNN*. 2018. arXiv: 1703.06870 [cs.CV].

[Jac+18] Dominic Jack et al. "Learning free-form deformations for 3D object reconstruction". In: *Asian Conference on Compurer Vision (ACCV)*. Springer. 2018, pp. 317–333.

[Kan+18] Angjoo Kanazawa et al. "End-to-end Recovery of Human Shape and Pose". In: *Computer Vision and Pattern Recognition (CVPR)*. 2018.

[Kur+18] Andrey Kurenkov et al. "DeformNet: Free-Form Deformation Network for 3D Shape Reconstruction from a Single Image". In: *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*. IEEE Computer Society, 2018, pp. 858–866. DOI: 10.1109/WACV.2018.00099. URL: https://doi.org/10.1109/WACV.2018.00099.

[OLV18] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation Learning with Contrastive Predictive Coding". In: *ArXiv* abs/1807.03748 (2018).

[Pav+18] Georgios Pavlakos et al. "Learning to Estimate 3D Human Pose and Shape from a Single Color Image". In: *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[Sun+18] Xingyuan Sun et al. "Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[Tul+18] Shubham Tulsiani et al. *Learning Shape Abstractions by Assembling Volumetric Primitives*. 2018. arXiv: 1612.00404 [cs.CV].

[Wan+18] Nanyang Wang et al. *Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images*. 2018. arXiv: 1804.01654 [cs.CV].

[Ave+19] Armen Avetisyan et al. "Scan2CAD: Learning CAD Model Alignment in RGB-D Scans". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[Dua+19] Kaiwen Duan et al. "CenterNet: Keypoint Triplets for Object Detection". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 6568–6577. DOI: 10.1109/ICCV.2019.00667.

[Geo+19] Georgios Georgakis et al. *Learning Local RGB-to-CAD Correspondences for Object Pose Estimation*. 2019. arXiv: 1811.07249 [cs.CV].

[Geo19] Justin Johnson Georgia Gkioxari Jitendra Malik. "Mesh R-CNN". In: *ICCV 2019* (2019).

[GRL19] Alexander Grabner, Peter M. Roth, and Vincent Lepetit. "Location Field Descriptors: Single Image 3D Model Retrieval in the Wild". In: *2019 International Conference on 3D Vision (3DV)*. 2019, pp. 583–593. DOI: 10.1109/3DV.2019.00070.

[Gro+19] Thibault Groueix et al. "Unsupervised cycle-consistent deformation for shape matching". In: *Symposium on Geometry Processing (SGP)*. 2019.

[Kuo+19] Weicheng Kuo et al. "ShapeMask: Learning to Segment Novel Objects by Refining Shape Priors". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 9206–9215. DOI: 10.1109/ICCV.2019.00930.

[LD19] Hei Law and Jia Deng. *CornerNet: Detecting Objects as Paired Keypoints*. 2019. arXiv: 1808.01244 [cs.CV].

[Mes+19] Lars Mescheder et al. "Occupancy Networks: Learning 3D Reconstruction in Function Space". In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[Mo+19a] Kaichun Mo et al. "PartNet: A Large-Scale Benchmark for Fine-Grained and Hierarchical Part-Level 3D Object Understanding". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[Mo+19b] Kaichun Mo et al. "StructureNet: Hierarchical Graph Networks for 3D Shape Generation". In: *ACM Transactions on Graphics (TOG), Siggraph Asia 2019* 38.6 (2019), Article 242.

[Nev+19] Davy Neven et al. *Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth*. 2019. arXiv: 1906.11109 [cs.CV].

[Pan+19] Junyi Pan et al. *Deep Mesh Reconstruction from Single RGB Images via Topology Modification Networks*. 2019. arXiv: 1909.00321 [cs.CV].

[Par+19] Jeong Joon Park et al. *DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation*. 2019. arXiv: 1901.05103 [cs.CV].

[Tat+19] Maxim Tatarchenko et al. "What Do Single-view 3D Reconstruction Networks Learn?" In: *CoRR* abs/1905.03678 (2019). arXiv: 1905.03678.

[Wu+19] Zhijie Wu et al. "SAGNet: Structure-aware Generative Network for 3D-Shape Modeling". In: *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019)* 38.4 (2019), 91:1–91:14.

[Xie+19] Haozhe Xie et al. "Pix2Vox: Context-aware 3D Reconstruction from Single and Multi-view Images". In: *ICCV*. 2019.

[Ave+20] Armen Avetisyan et al. "SceneCAD: Predicting Object Alignments and Layouts in RGB-D Scans". In: *The European Conference on Computer Vision (ECCV)*. Aug. 2020.

[CTZ20] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. *BSP-Net: Generating Compact Meshes via Binary Space Partitioning*. 2020. arXiv: 1911.06971 [cs.CV].

[Den+20] Boyang Deng et al. "CvxNet: Learnable Convex Decomposition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[Gra+20] Alexander Grabner et al. "Geometric Correspondence Fields: Learned Differentiable Rendering for 3D Pose Refinement in the Wild". English. In: *Computer Vision – ECCV 2020*. Lecture Notes in Computer Science. 16th European Conference on Computer Vision : ECCV 2020, ECCV 2020 ; Conference date: 23-08-2020 Through 28-08-2020. Springer, 2020, pp. 102–119. ISBN: 978-3-030-58516-7. DOI: 10.1007/978-3-030-58517-4_7.

[Jak+20] Tomas Jakab et al. "KeypointDeformer: Unsupervised 3D Keypoint Discovery for Shape Control". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[Jia+20] Chiyu Jiang et al. "ShapeFlow: Learnable Deformations Among 3D Shapes". In: *Advances in Neural Information Processing Systems*. 2020.

[Kir+20] Alexander Kirillov et al. "PointRend: Image Segmentation As Rendering". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[Kuo+20] Weicheng Kuo et al. "Mask2CAD: 3D Shape Prediction by Learning to Segment and Retrieve". In: *ECCV*. 2020.

[Lev+20] Jake Levinson et al. "An Analysis of SVD for Deep Rotation Estimation". In: *CoRR* abs/2006.14616 (2020). arXiv: 2006.14616. URL: https://arxiv.org/abs/2006.14616.

[Man+20] Kevis-Kokitsi Maninis et al. "Vid2CAD: CAD Model Alignment using Multi-View Constraints from Videos". In: *arXiv* (2020).

[Mil+20] Ben Mildenhall et al. *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis*. 2020. arXiv: 2003.08934 [cs.CV].

[Nie+20] Yinyu Nie et al. *Total3DUnderstanding: Joint Layout, Object Pose and Mesh Reconstruction for Indoor Scenes from a Single Image*. 2020. arXiv: 2002.12212 [cs.CV].

[PBF20] Stefan Popov, Pablo Bauszat, and Vittorio Ferrari. "CoReNet: Coherent 3D Scene Reconstruction from a Single RGB Image". In: *Computer Vision – ECCV 2020*. 2020. DOI: 10.1007/978-3-030-58536-5_22.

[Sun+20] Minhyuk Sung et al. "DeformSyncNet: Deformation Transfer via Synchronized Shape Deformation Spaces". In: *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia)* (2020).

[Uy+20] Mikaela Angelina Uy et al. "Deformation-Aware 3D Model Embedding and Retrival". In: *European Conference on Computer Vision (ECCV)*. 2020.

[Yif+20] Wang Yifan et al. *Neural Cages for Detail-Preserving 3D Deformations*. 2020. arXiv: 1912.06395 [cs.GR].

[Yu+20] Alex Yu et al. *pixelNeRF: Neural Radiance Fields from One or Few Images*. 2020. arXiv: 2012.02190 [cs.CV].

[ZKG20] Wei Zeng, Sezer Karaoglu, and Theo Gevers. *Inferring Point Clouds from Single Monocular Images by Depth Intermediation*. 2020. arXiv: 1812.01402 [cs.CV].

[Dos+21] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *ICLR* (2021).

[Eng+21] Francis Engelmann et al. *From Points to Multi-Object 3D Reconstruction*. 2021. arXiv: 2012.11575 [cs.CV].

[Gu+21] Geonmo Gu et al. *Towards Real-time and Light-weight Line Segment Detection*. 2021. arXiv: 2106.00186 [cs.CV].

[Liu+21] Ze Liu et al. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows". In: *arXiv preprint arXiv:2103.14030* (2021).

[Su+21] Yongzhi Su et al. "SynPo-NetâAccurate and Fast CNN-Based 6DoF Object Pose Estimation Using Synthetic Training". In: *Sensors* 21.1 (2021). ISSN: 1424-8220. DOI: 10.3390/s21010300. URL: https://www.mdpi.com/1424-8220/21/1/300.

[Uy+21] Mikaela Angelina Uy et al. "Joint Learning of 3D Shape Retrieval and Deformation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[Vak+21] Alexander Vakhitov et al. "Uncertainty-Aware Camera Pose Estimation From Points and Lines". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4659–4668.

[Zim21] Jan Zimoch. "Multimodal Scene Coordinate Regression: A Novel, Probabilistic Approach to Semantic Re-Localisation in Ambiguous Environments". Master's Thesis. University of Cambridge, 2021.

Appendix A

Similarity of Train and Test CAD Models

The following provides additional information about the Pix3D[Sun+18] dataset for which [Geo19] introduced two data splits. For the S1 split the 10,069 images are randomly split into 7539 train images and 2530 test images. Under this split all CAD models are seen during training. For the S2 split train and test images are split such that the test images contain CAD models that were not present in the training images. The challenge is therefore to construct a system that given an input image is able to retrieve an unseen CAD model and precisely predict its pose.

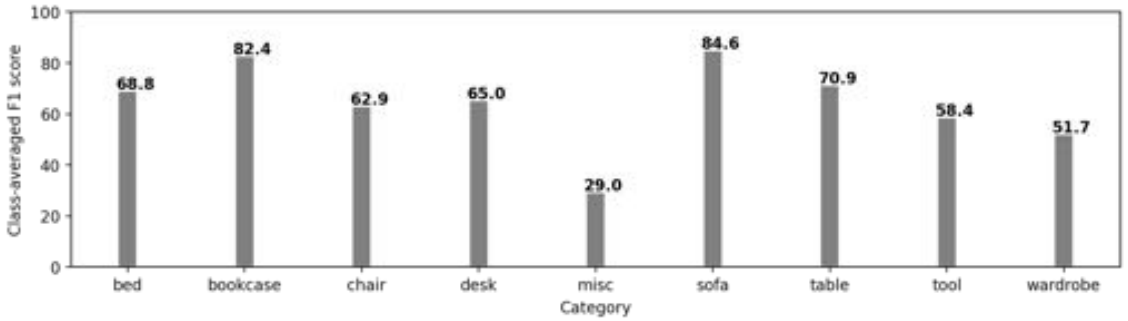


Figure A.1: Grey bars provide an indicator for the similarity between CAD models seen during training and unseen CAD models used for testing under the S2 split of Pix3d[Sun+18]. Quantitatively grey bars show the class average when the F1 score is computed between every unseen CAD model and its closest matching CAD model (in terms of the F1 score) from the seen ones.

We have demonstrated in our main work that the geometric approach that we follow is more accurate compared to directly predicting object poses[Kuo+20] on the S2 split of Pix3D[Sun+18]. Further, we will show here that the good performance

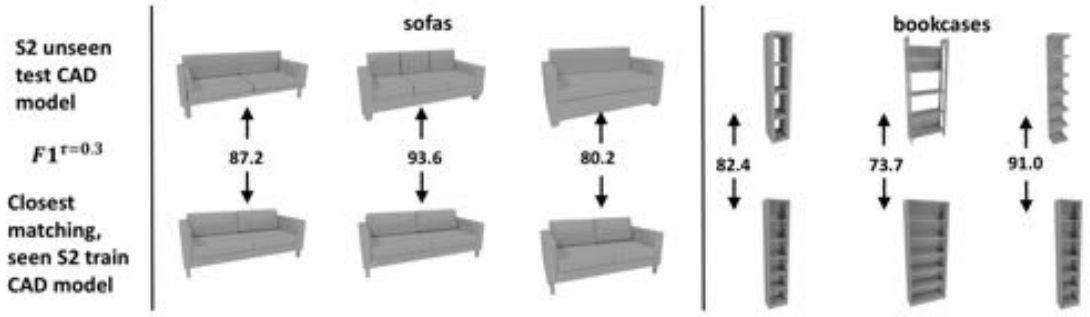


Figure A.2: Visualisation of selected test CAD models and their closest matching train CAD models in terms of the F1 score. We note the strong similarity (both visually and in terms of the F1 score) between sofas in the test split and sofas from the train split. While bookcases in the test split also have close matching CAD models in the train split in terms of the F1 score, they differ significantly in their visual appearance. This increases the difficulty for retrieval at test time and explains the poor performance of [Kuo+20] on bookcases compared to sofas.

[Kuo+20] achieves on sofas does not require it to retrieve unseen CAD models as for every unseen CAD model in the test images there is a closely fitting CAD model among the seen training CAD models. We quantify this by computing the F1 score at $\tau = 0.3$ between unseen test CAD models and their closest matching CAD models (in terms of F1 score) from the seen ones. We perform this calculation for all unseen CAD models and compute the mean to obtain class averages. These are plotted in gray in Figure A.1. Note here that the averaged best-possible F1 score for sofas is 84.6 which is exceptionally high compared to other class averages. This strong similarity (see Figure A.2 for selected test CAD models and their closest-matching CAD models from the train set) allows [Kuo+20] to make accurate shape predictions without retrieving unseen CAD models. We also note that [Kuo+20] performs poorly on bookcases despite a high class-averaged F1 score between test CAD models and their closest matching train CAD models. The reason for this is that while good candidate CAD models exist in the seen train CAD models in terms of the F1 score, their different visual appearance (see right side Figure A.2) makes them difficult to retrieve at test time.