Inter and Intra Image Relationships in 3D Space

Nilesh Kulkarni

May 20, 2019

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Thesis Committee:
Abhinav Gupta, *Carnegie Mellon University*, Chair
Martial Hebert, *Carnegie Mellon University*
Gunnar Atli Sigurdsson, *Carnegie Mellon University*

Tech Report:- CMU-RI-TR-19-31

Thesis proposal submitted in partial fulfillment of the requirements for the degree of Masters in Robotics

Abstract

Relationships are at the core of understanding images. A lot of progress in computer vision is due to the ability of convolutional neural networks to model relationships between pixels in images across our massive data-sets. Convolutional neural networks bake in relationships between pixels both within images and across image collections to perform better on tasks like image segmentation, object detection, etc.. Relationship can exists in the pixel, concept and object space which is a huge spectrum. We show that different kinds of relationships are relevant for different tasks. This thesis explores various other ways to model "*inter and intra image relationships*" to help with the tasks of semantic correspondences, and 3D reconstruction.

In this thesis we explore the following two tasks a) semantic correspondence between images from the same category of objects where we show how relationships can be inherently learned and how geometric constraints helps us learn global meaningful relationships and b) 3D reconstruction from a single image where we explicitly factor relationships in 3D to help with the task of 3D reconstruction to improve the overall performance across multiple datasets.

Acknowledgements

I have learned from so many people before and after joining CMU. This work has been only possible due to the motivation and teaching from all my teachers and friends.

I am immensely grateful to my advisor Abhinav Gupta for the unwavering support, his insights, ideas, perspectives about research and foremost providing me with freedom to choose research problems at will! He has been very supportive and allowed me to learn a lot from my failures. I thank you for having the confidence in me. I have enjoyed all the house parties we had and I am grateful for having such a great lab environment to work in!

My collaborators Ishan Misra and Shubham Tulsiani have been instrumental and of immense help with these works. I would like to thank Ishan for his constant support, motivation and helping me make the right choices during the course of the masters at CMU and making sure I was well-directed. I also want to thank Shubham for his continued support in pursing different aspects of research, listening to my various idea and giving continuous feedback at any hour of the day! I have learned a lot from both of you! Thank you for teaching me so much.

I would like to thank my committee members Martial Hebert and Gunnar Sigurdsson for their comments on my presentations, insights, and for taking time to give valuable feedback.

My friends, labmates and collegaues at CMU who gave me a great environment to complete my masters. A special shout out to - Akshat Agarwal, Shubham Agrawal, Aayush Bansal, Divyam Bansal, Tao Chen, Achal Dave, Victoria Dean, Dhiraj Gandhi, Rohit Girdhar, Saurabh Gupta, Helen Jiang, Rawal Kirodhkar, Sumit Kumar, Aashi Manglik, Kenny Marino, Akshita Mittal, Gaurav Mittal, Adithya Murali, Siva Mynepalli, Ishan Nigam, Gaurav Pathak, Lerrel Pinto, Senthil Purushwalkam, Samantha Powers, Dinesh Reddy, Pratyusha Sharma, Xiaolong Wang, Tian Ye, Judy Ye, Wenxuan Zhou

I would like to thank my parents and sister for allowing me to pursue grad school and supporting me all these years. I would have never been here if not for your unconditional love, energy and support you provide me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Relationships and context play an important role in understanding the visual world. Our intuition is to exploit relationships in our world that stem from its three-dimensional nature. We show how relationships can be learned and are also useful to in certain tasks. The works presented in thesis relies on exploiting 3D structure which includes relationships between objects, camera, and the pixels. In the first work, we explore the camera and image relationships and in the second one, we explore spatial relationships between objects.

We explore the task of Canonical Surface Mapping (CSM). Specifically, given an image, we learn to map pixels on the object to their corresponding locations on an abstract 3D model of the category. But how do we learn such mapping? A supervised approach would require extensive manual labeling which is not scalable beyond a few hand-picked categories. Our key insight is that the CSM task (pixel to 3D), when combined with 3D projection (3D to pixel), completes a cycle. Hence, we can exploit a geometric cycle consistency loss, thereby allowing us to forgo dense manual supervision. Our approach allows us to train a CSM model for a diverse set of classes, without sparse or dense keypoint annotation, by leveraging only foreground mask labels for training. We show that our predictions also allow us to infer dense correspondence between two images, and compare the performance of our approach against several methods that predict correspondence by leveraging the varying amount of supervision.

Next, we propose an approach to predict the 3D shape and pose for the objects present in a scene. Existing learning based methods that pursue this goal make independent predictions per object and do not leverage the relationships amongst them. We argue that reasoning about these relationships is crucial, and present an approach to incorporate these in a 3D prediction framework. In addition to independent per-object predictions, we predict pairwise relations in the form of the relative 3D pose and demonstrate that these can be easily incorporated to improve object level estimates. We report performance across different datasets (SUNCG, NYUv2), and show that our approach significantly improves over independent prediction approaches while also outperforming alternate implicit reasoning methods. The relevant publication list referred to in this thesis is as follows:

1. Chapter **??**: Canonical Surface Mapping via Geometric Cycle Consistency, *Under Submission* [**?**]
2. Chapter **??**: 3D-RelNet: Joint Object and Relational Network for 3D Prediction, *Under Submission* [**?**]

Chapter 2

Canonical Surface Mapping via Geometric Cycle Consistency

2.1 Introduction

Plato famously remarked that while there are many cups in the world, there is only one 'idea' of a cup. Any particular instance of a category can thus be understood via its relationship to this platonic ideal. As an illustration, consider an image of a bird in Figure **??**. When we humans see this image, we can not only identify and segment the bird but also go further and even map pixels to an abstract 3D representation of the category. This task of mapping pixels in an image to locations on an abstract 3D model (which we henceforth call **canonical surface mapping**) is generalization and densification of keypoint estimation and is key towards rich understanding of objects. But how do we learn to do this task? What is the right data, supervision or models to achieve dense rich understanding of objects?

One way to learn the canonical surface mapping task is to collect large-scale labeled data. Specifically, we can label hundreds or thousands of keypoints per image for thousands of images. As each keypoint location defines which pixel corresponds to a specific location on the 3D surface, this approach of manually labeling the keypoints can provide dense supervision for learning. This approach has in fact been shown to be quite successful for specific categories such as humans [**?**]. But of course collecting such labeled data requires enormous manual labeling effort, making it difficult to scale to generic categories.

Is there an alternative supervisory signal that can allow one to learn without reliance on such labelled data? Interestingly, we note that this task of canonical surface mapping is an inverse graphics task. Any such mapping is constrained by the geometry operating on the underlying 3D, and any predicted mapping should also respect this structure. In particular, for the pixels that belong to the object given by the object mask, the CSM function maps these pixels onto the 3D shape. These points on the 3D shape, when projected back using (known/predicted) camera, should map back to the same pixels. Our key insight is that one can complete the cycle (pixels $\rightarrow$ 3D $\rightarrow$ pixels) and use the consistency loss as an objective. The gradients from the loss can be propagated back to the CSM function prediction function, thereby allowing us to learn this mapping without reliance on strong forms of supervision.

In this paper, we present an approach to learn the task of canonical surface mapping from the set of images belonging to semantic category, their input masks and an abstract 3D model which represents the semantic category. Additionally, we show that predicting a canonical surface mapping for

images allows us to infer dense correspondence across images of a category, and our approach enables recovery of dense correspondences without any correspondence supervision! In comparison to approaches that use dense supervision for this task [**?**], or approaches that leverage keypoints for the related tasks of semantic correspondence [**?**], or 3D reconstruction [**?**], this is significant decrease in supervision. This allows us to train for CSM-model for diverse set of classes: birds, zebras, cars and more (See figure **??**). We believe our approach can pave the way for large-scale internet-driven 3D understanding and correspondence inference since both semantic imagesets and masks are easy to obtain (and automatic approaches can be used as well).

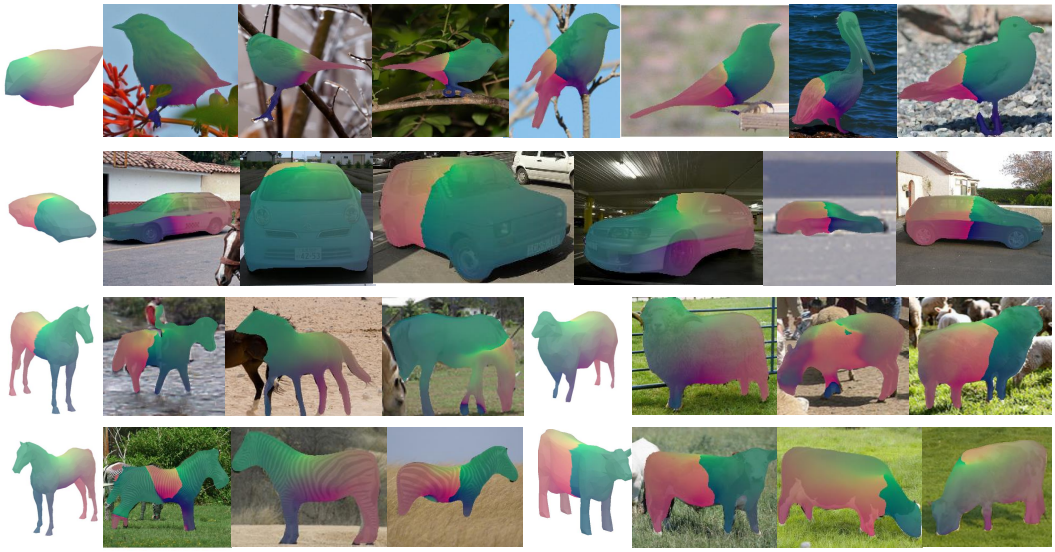

Figure 2.1: We study the task of Canonical Surface Mapping. This task is a generalization of keypoint estimation and involves mapping pixels to canonical 3D models. We learn CSM prediction without requiring correspondence annotations, by instead using geometric cycle consistency as supervision. This allows us to train CSM prediction for diverse classes, including rigid and non-rigid objects.

2.2 Related Work

Dense Semantic Correspondences. A fundamental task that is equivalent to pursuing canonical surface mapping is that of inferring dense semantic correspondence – given two images, the goal is to predict for each pixel in the former, the corresponding pixel in the latter. Methods prior to the recent resurgence of deep learning [**?**, **?**] demonstrated that matching using features such as SIFT could allow recovering correspondence across instances, and later work showed similar results using CNN features [**?**, **?**]. While these generic features allow recovering correspondence, learning specifically for the task using annotated data can improve results [**?**]. However, collecting such annotation can be tedious, so several approaches have attempted to relax the supervision for learning correspondence. Among these, a common paradigm is to learn correspondence by self-supervision, where random perturbations of images are used as training pairs. This allows predicting parametric warping [**?**,**?**,**?**] to relate images, or learn equivariant embeddings [**?**] for matching. However, the these methods are fundamentally restricted to training data of the same instance, with no change in the visible content, thereby limiting the performance for different instances with viewpoint changes. An alternate form of supervision can come via synthetic data, where synthetic image pairs rendered using the same

pose as a real image pair, can help learn a correspondence function between real images that is cycle-consistent [**?**]. However, this approach relies on availability of large-scale synthetic data and known pose for real images to generate the supervisory signal, and we show that both these requirements can be relaxed.

Learning Invariant Representations. Our work is broadly related to methods that learn pixel embeddings invariant to certain transforms. These approaches leverage tracking to obtain correspondence labels, and learn representations invariant to viewpoint transformation [**?**, **?**] or motion [**?**]. Similar to self-supervised correspondence approaches, these are also limited to training using observations of the same instance, and do not generalize well across instances. While our canonical surface mapping is also a pixel-wise embedding invariant to certain transforms, it has a specific geometric meaning *i.e*. correspondence to a 3D surface, and leveraging this is what allows learning without the correspondence supervision.

Category-Specific 3D Reconstruction. A related line of work pursued in the community is that of reconstructing the instances in a category using using category-specific deformable models. Dating back to the seminal work of Blanz & Vetter [**?**], who operationalized D'Arcy Thompson's insights into the manifold of forms [**?**], morphable 3D models have been used to model faces [**?**], hands [**?**,**?**], humans [**?**, **?**] and other generic classes [**?**, **?**, **?**, **?**].
In conjunction with known/predicted camera parameters, this representation also allows one to extract a pixelwise canonical mapping. However, these methods often rely on 3D training data to infer this representation. Even approaches that relax this supervision [**?**, **?**, **?**] crucially rely on (sparse or dense) 2D keypoint annotations during training. In contrast, we show that learning a canonical surface mapping is feasible even without such supervision. Further, we demonstrate that directly learning the mapping function leads to more accurate results than obtaining these via an intermediate 3D estimate.

Consistency as Meta-Supervision. Ours is not the only task where acquiring direct supervision is often infeasible, and the idea of leveraging some form of consistency to overcome this hurdle has been explored in several domains. Recent volumetric reconstruction [**?**, **?**, **?**, **?**] or depth prediction [**?**, **?**, **?**] approaches use geometric consistency between the predicted 3D and available views as supervision. Similarly, the notion that when learning some transformations, their composition often respects a cyclical structure has been used for image generation [**?**, **?**], correspondence estimation [**?**, **?**] *etc*. In our setup, we also observe that the approach of using consistency as meta-supervision allows bypassing supervision. We do so by leveraging insights related to both, geometry and cycle consistency – given a surface mapping, there is a geometrically defined inverse transform with which the canonical surface mapping predictions should be cycle-consistent.

2.3 Approach

Given an image, our goal is to infer for each pixel on the object, its mapping onto a given canonical template shape of the category. We do so by learning a parametrized CNN f_θ, which predicts a pixelwise canonical surface mapping (CSM) given an input image. We show that our method, while only relying on foreground masks as supervision, can learn to map pixels to the given category-level template shape. Our key insight is that this mapping function we aim to learn has a geometric structure that should be respected by the predictions. We operationalize this insight, and learn a CSM predictor using a geometric cycle consistency loss, thereby allowing us to bypass the need for supervision in the form of annotated (sparse or dense) keypoints.
We first present in Section **??** our training setup in a scenario where the camera pose for each

training image is given. We then show how we can relax this requirement of known camera in Section **??**. Learning a CSM predictor implicitly allows us to capture the correspondence across instances, and we describe in Section **??** the procedure to recover dense semantic correspondence given two images.

2.3.1 Preliminaries

Surface Parametrization. The template shapes we learn mappings to are in fact two-dimensional surfaces in 3D space. The surface S of the template shape can therefore be parametrized via two parameters $u \in (0, 1)$ and $v \in (0, 1)$ (or equivalently a 2D vector $\mathbf{u}$). This parametrization implies that we can obtain a mapping ϕ such that $\phi(\mathbf{u})$ represents a unique point on the surface S.

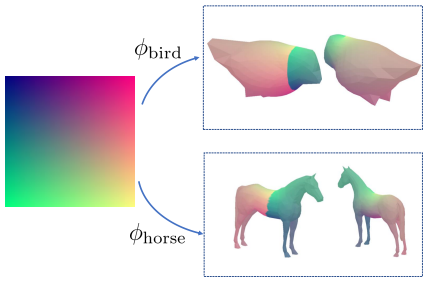

Figure 2.2: Surface Parametrization. We show the mapping from (u, v) space to the surface of the 3D model for two categories.

While there are several ways to construct such a mapping, one intuitive way is to consider $\mathbf{u}$ to represent the polar angles to parametrize points on the surface of a hollow sphere, which can be mapped to a surface S by pushing it inward [**?**]. Given a template shape with a surface S, we use this approach to obtain the parametrization ϕ. We show some visualizations in Figure **??** for the mapping from a 2D square to template 3D shapes for two categories.

Canonical Surface Mapping. A canonical surface mapping C for an image I is a mapping from pixels onto the template 3D shape. Given a pixel $\mathbf{p} \equiv (x, y)$, $C[\mathbf{p}]$ represents the corresponding point on the surface. As the surface has a two-dimensional parametrization, C is equivalently an image of the same size as I, with a two-channel value at each pixel. Our parametrized CNN f_θ that predicts this mapping from an input image, therefore learns a per-pixel prediction task – given an RGB input image, it outputs a 2 dimensional vector for each pixel.

Camera Projection. We model the camera as a weak perspective (scaled orthographic) transformation. We represent the camera for every image I as π, parameterized by the scale $s \in \mathcal{R}$, translation $t \in \mathcal{R}^2$ and rotation $q \in \mathcal{R}^4$ as a (normalized) quaternion. We denote by $\pi(P)$ as the projection of a point P to the image coordinate frame using the camera parameters $\pi \equiv (s, t, r)$.

2.3.2 Learning via Geometric Cycle Consistency

We aim to learn a per-pixel predictor f_θ that outputs a canonical surface mapping given an input image I. We present an approach to do so using only foreground masks as supervision. However, for simplicity, we first describe here how we can learn this CSM predictor assuming known camera parameters for each training image, and relax this requirement in Section **??**.

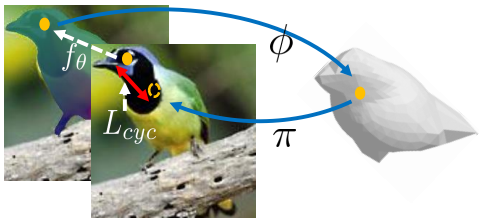

Figure 2.3: Geometric Cycle Consistency Loss. A pixel mapped to $\mathbf{u}$ by CSM function f_θ gets mapped onto the 3D template via ϕ. Our loss enforces that this 3D point, when projected back via the camera π, should map back to the pixel.

Our approach is to derive learning signal from the geometric nature of this task. In particular, as the 3D shapes underlying instances of a category are often similar (and therefore similar to the template shape), a pixel-wise mapping onto the 3D surface should be (approximately) cycle-consistent under reprojection. We capture this constraint via a geometric cycle consistency loss. This loss, in conjunction with an objective that allows the prediction to respect certain visibility constraints, allows us to learn f_θ.

Geometric Cycle Consistency Loss. Given an image I with associated camera π and foreground mask I_f, we wish to enforce that the predicted canonical surface mapping $C \equiv f_\theta(I)$, respects the underlying geometric structure. Concretely, as the instances across a category bear resemblance to the template shape, given a pixel $\mathbf{p}$ on the object foreground, we would expect that its corresponding point on the 3D surface $\phi(C[\mathbf{p}])$ to (approximately) project back under the camera π. We define a geometric consistency loss (see Figure **??**) that penalizes this inconsistency for all foreground pixels, thereby encouraging the network to learn pixel $\rightarrow$ 3D mapping functions that are cycle-consistent under the 3D $\rightarrow$ pixel reprojection.

$$L_{cyc} = \sum_{\mathbf{p} \in I_f} \|\pi(\phi(C[\mathbf{p}])) - \mathbf{p}\|_2^2 \tag{2.1}$$

Incorporating Visibility Constraints. Enforcing that the pixels when lifted to 3D, project back to the same location is desirable, but not a sufficient condition. As an illustration, for a front facing bird, both the beak and tail project at similar locations, but only the former would be visible. This implies that points on the surface that are self-occluded under π can also result in minimizing L_{cyc}. Our solution is to discourage f_θ from predicting $\mathbf{u}$ values that map to self-occluded regions under camera π.

We note that if a point on the 3D shape is self-occluded under a camera π, its z-coordinate in camera frame is larger than the visible depth at the corresponding pixel. We use Neural Mesh Renderer (NMR) [**?**] to render a depth map D_π for the tempalte shape S under camera π, and define a visibility loss for each pixel ı by checking if the z-coordinate (say $z_\mathbf{p}$) of its corresponding point $\phi(C[\mathbf{p}])$ on the 3D shape, when projected under π, has a larger z-coordinate.

$$L_{vis} = \sum_{p \in I_f} \max(0, z_\mathbf{p} - D_\pi[\mathbf{p}]) \tag{2.2}$$

Network Details. We implement f_θ as a network with UNet [**?**] style architecture. This network takes as input an image of size 256 x 256 and outputs a unit vector per pixel representing a point on

surface of sphere parameterized by (u, v). We train our network to minimize the cycle-consistency and visibility objectives:

$$L_{\text{consistency}} = L_{vis} + L_{cyc} \tag{2.3}$$

Even though we do not have direct supervision for the mappings, as we train a shared predictor across instances, the explicit priors for geometric consistency, and the implicit inductive biases in CNNs for spatial equivariance are sufficient for us to learn a meaningful predictor.

Foreground Mask Prediction. While the training procedure described above encourages cycle-consistent predictions at pixels belonging to the object, the learned CNN f_θ also predicts some (possibly spurious) values at other pixels. To allow us to ignore these background pixels for inferring correspondence (see Section. **??**), as well as for generating visualizations, we train an additional per-pixel mask predictor using standard cross-entropy loss L_{fg} against the ground-truth masks. To do so, we simply modify f_θ to yield an additional per-pixel foreground probability as output.

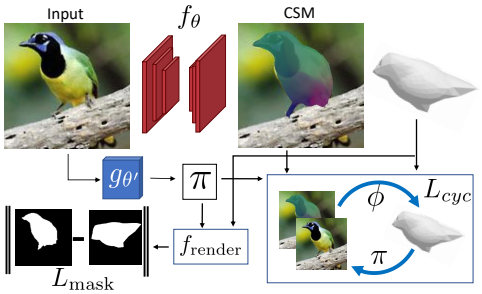

Figure 2.4: Overview of Training Procedure. We train a network to predict, for each pixel on the foreground, its mapping to the canonical shape. We also jointly learn to predict camera pose, and the geometric cycle-consistency loss L_{cyc} alongwith foreground supervision, provides learning signal to train our system.

2.3.3 Learning without Pose Supervision

We have presented our approach to learn a canonical surface mapping predictor f_θ assuming known cameras π for each training image. We note that our training objective is also differentiable w.r.t. the camera parameters, and we can therefore simply use predicted cameras instead of known cameras, and jointly learn pose and CSM prediction. This joint training can allow us to bypass the requirement of even camera supervision, and learn CSM prediction using only foreground mask annotations and a given template shape.

We therefore learn an additional camera-prediction CNN $g_{\theta'}$, and use the predicted cameras to learn the CSM predictor via the geometric consistency training objectives. However, to overcome certain trivial solutions, we also add a mask reprojection error, and following [**?**, **?**] use a multi-hypothesis camera predictor to avoid local minima. Our overall training setup is depicted in Figure **??**.

Mask Re-projection Loss. If the only learning objective comprises of the self-consistency between camera predictions and the predicted CSMs, the networks can learn some trivial solutions *e.g.* always predict a 'frontal' camera and corresponding CSM. To avoid this we enforce that the the template shape, when viewed under a predicted camera π, should approximately match the known foreground image I_{fg}. To implement this loss, we use (NMR) [**?**] to obtain a differentiable render f_{render}, that given the template shape S and a camera π, renders a mask. This additional mask reprojection loss,

which allows us to circumvent the mentioned trivial solutions, can be denoted as follows:

$$L_{\text{mask}} = \|f_{\text{render}}(S, \pi) - I_f\|^2 \tag{2.4}$$

Multi-Hypothesis Pose Prediction. Instead of predicting a single camera $\pi \equiv g_{\theta'}(I)$, we follow previous methods [**?**, **?**] and predict multiple hypotheses to overcome local minima. Our pose predictor outputs $\{(\pi_i, c_i)\} \equiv g_{\theta'}(I)$ - a set of $N_c = 8$ pose hypotheses π_i, each with an associated probability c_i. We initialize the camera predictor $g_{\theta'}$ using a pre-trained ResNet-18 network [**?**].

Overall Training Objective. As our pose predictor yields multiple pose hypotheses π_i, each with an associated probability c_i, we can train our networks by minimizing the expected loss. We denote by $L^i_{cyc}, L^i_{vis}, L^i_{mask}$ the corresponding losses under the camera prediction π_i. In addition to minimizing the expected loss over these terms, we also use an additional diversity prior L_{div} to encourage diverse pose hypotheses (see appendix for details). The overall training objective using these, is:

$$L_{tot} = L_{div}(g_{\theta'}(I)) + \sum_{i=1}^{N_c} c_i(L^i_{cyc} + L^i_{vis} + L^i_{mask}) \tag{2.5}$$

This framework allows us to learn the canonical surface mapping function f_θ via geometric cycle consistency, using only foreground mask annotations in addition to the given template shape. Once the network f_θ is learned, we can infer a canonical surface map from any unannotated image.

2.3.4 Dense Correspondences via CSM

We described an approach for predicting canonical surface mappings without relying on pose or keypoint annotations. This allows us to infer dense semantic correspondences given two images of the same semantic object category, because if pixels across images correspond, they should get mapped to the same region on the canonical surface. Given a (source, target) image pair (I_s, I_t), let us denote by $(C_s, C_t, I^s_{fg}, I^t_{fg})$ the corresponding predicted canonical surface mappings and foreground masks. Given these predictions, for any pixel $\mathbf{p}_s$ on I_s, we can infer its corresponding pixel $T_{s\to t}[\mathbf{p}_s]$ on I_t by searching for the (foreground) pixel that maps closest to $\phi(C_s[\mathbf{p}_s])$.

$$T_{s\to t}[\mathbf{p}_s] = \arg\min_{\mathbf{p}_t \in I^t_{fg}} \|\phi(C_s[\mathbf{p}_s]) - \phi(C_t[\mathbf{p}_t])\| \tag{2.6}$$

Not only does our approach allow us to predict correspondences for pixels between two images, it *also allows us to infer regions of non-correspondence i.e.* pixels in source image for which correspondences in the target image do not exist (*e.g.* most pixels between a left and right facing bird do not correspond). We can infer these by simply denoting pixels for which the minimum distance in Eq. **??** is above a certain threshold as not having a correspondence in the target image. This ability to infer non-correspondence is particularly challenging for self-supervised methods that generate data via random warping [**?**, **?**, **?**] as the training pairs for these never have non-corresponding regions.

2.4 Experiments

Our approach allows us to predict canonical surface mappings across generic categories. However, due to lack of annotation for the task, which is in fact our motivation for learning without supervision, it is difficult to directly evaluate the predictions. Instead, as our approach also allows us to

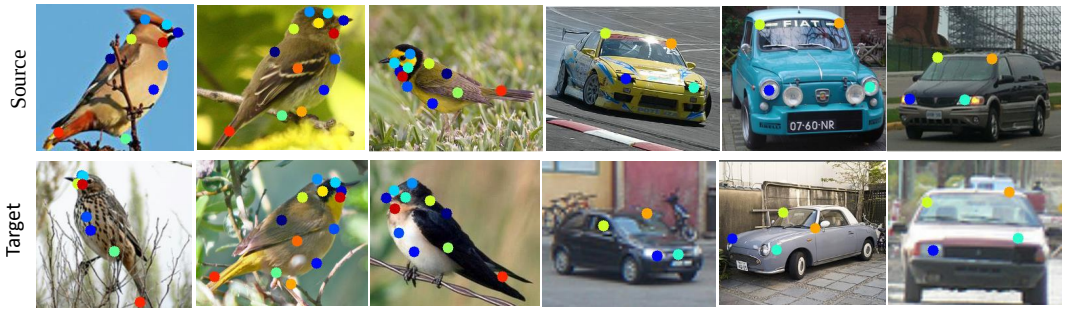

Figure 2.5: Keypoint transfer results. We show the quality of dense correspondence results by transferring ground-truth keypoints from **source** images in the top-row to **target** images in the bottom-row. It is interesting to note that method is able to transfer keypoints despite significant changes in the viewpoint.

recover correspondences across any two images (Section **??**), we can evaluate these using the task of keypoint transfer. This is a well-studied task by approaches that learn semantic correspondence, and we report comparisons to baselines that leverage varying degree of supervision while training. We first report these comparisons in Section **??**, and then present results for additional generic categories (e.g. horses, sheep, cows) in Section **??**, using Imagenet images with automatically obtained segmentation masks.

2.4.1 Evaluation via Keypoint Transfer

We use our learned CSM prediction models for the task of keypoint transfer – given a source and target image pair, where the source image has some annotated keypoints, the goal is to predict the location of these keypoints in the target image. We first describe the datasets used to train our model, and then briefly survey the various baselines we compare to and then present the evaluation results.

Experimental Setup

Datasets. We use bird images from the CUB-200-2011 [**?**] and the car images from PASCAL3D+ [**?**] dataset for quantitative evaluation. CUB-200-2011 contains 6000 training and test images with 200 different species. Each bird has 14 annotated keypoints, a segmentation mask, and a bounding box. Note that we only use keypoint annotation at test time to evaluate our method on the task of dense correspondence as described earlier. We also evaluate on the car category PASCAL3D+ which has over 6000 training and test images with 12 keypoint annotations per instance. We downloaded a freely available mesh from [**?**] to serve as a bird template shape, used an average of 10 Shapenet [**?**] models to obtain a template shape for cars.

Baselines. We report comparisons to several methods that leverage varying amount of supervision for learning:

Category Specific Mesh Reconstruction (CMR) [**?**] learns to reconstruct the 3D shape and predict pose for a given instance, but relies on training time supervision of known keypoint locations and segmentation masks. Since a common morphable model is used across a category, we can compute the implied surface mappings via computing for each pixel, the coordinate of the mean shape that is rendered at its location (or nearest location in case of imperfect projection). We can then infer correspondences as in Section **??**.

Zhou et al. [**?**] exploit a large collection of 3D synthetic models to learn dense correspondence via cycle-consistency. During training, they crucially rely on pose supervision (from PASCAL 3D+), as each cycle consists of synthetic images rendered from the same view as the real image pair. Their method outputs dense correspondences in the form of a per-pixel flow, and infers non-correspondence using a 'matchability' score.

Dense Equivariance (DE) [**?**] is a self-supervised method to learn correspondences, and does not require any pose or keypoint annotations. We re-implement this baseline such that it can exploit the annotations for object masks (see appendix for details). DE learns a per-pixel feature vector, and enforces corresponding pixels to have a similar feature. The supervision for correspondences is obtained via applying known in-plane random warps to images. During inference, we can recover the correspondence for a source pixel by searching for the most similar feature in the target image.

VGG Transfer. Long *et al*. [**?**] showed that it is feasible to recover correspondences via generic learned features for image classification. Therefore, instead of leveraging specifically learned feature as in (DE), we use features from a pre-trained VGG network [**?**] (we found these features to perform better than AlexNet used by Long *et al*. [**?**]).

Evaluation Metrics

We evaluate the various methods on two metrics: a) Percentage of Correct Keypoints (PCK), and b) Keypoint Transfer AP (APK). We use two separate metrics, because while the PCK metric evaluates the accuracy of keypoint transfer for keypoints that are visible in both, source and target image, it does not disambiguate if an approach can infer that a particular source keypoint does not correspond to any pixel on the target. So therefore, while PCK lets us evaluate correspondence accuracy, the APK metric also lets us measure accuracy at inferring *non-correspondence*.

Percentage of Correct Keypoints (PCK): Given a (source, target) image pair with keypoint annotations on the source, each method predicts a single estimate for the corresponding location in the target image. The PCK metric reports the mean accuracy of keypoint predictions across keypoints that are common across pairs. A prediction is considered correct only when the predicted location lies within $\alpha * \max(h, w)$ radius around the ground truth annotation for the transfer. We report results with $\alpha = 0.1$, and h, w refer to height and width of the image to which the keypoints were transferred.

Keypoint Transfer AP (APK): In addition to predicting a location in target image for each each keypoint in source image, this metric requires a confidence in the estimate. Ideally, if a source keypoint does not correspond in a target image, the corresponding predicted confidence should be low, whereas it should be high in case of a keypoint visible in both. Our approach and CMR [**?**] can rely on the (inverse) distance on the template/mean shape as a confidence measure. Zhou *et al*. [**?**] produce a 'matchability' score, and the feature based methods 'DE' [**?**] and 'VGG transfer' [**?**] can leverage feature similarity as confidence.

Given these predictions, we vary the confidence thresholds, and plot 'Transfer Precision' vs 'Transfer Recall' and report the area under the curve as the AP. 'Transfer Recall' measures the fraction of correspondences in the ground-truth that have been recovered above the threshold (at the lowest confidence threshold, this value is similar to PCK). 'Transfer Precision' measures the fraction of correspondences above the threshold that are correct (a prediction for a non-corresponding keypoint is always deemed incorrect). For a high precision, a method should predict low confidence scores for non-corresponding keypoints. We explain these metrics in more detail in the appendix.

Results

In addition to reporting the performance of our method, without any pose supervision, we also evaluate our approach when using pose supervision (denoted as 'CSM w/Pose') to better compare to baselines that use similar [**?**] or more [**?**] annotations. However, note that all results visualization in the paper are in a setting *without* known pose. We report the PCK and APK results in Table **??**, and observe that our approach performs better than the alternatives. We also show the Transfer AP plots in Figure **??**, and note large the relative performance boost (in particular over the self-supervised method [**?**]), indicating that our approach, in addition to inferring correspondences when they exist, can realize when regions do not correspond. We also visualize some qualitative results for keypoint transfer in Figure **??**.

2.4.2 Learning from Unannotated Image Collections

As our method does not require keypoint supervision during training, we can apply it to learn canonical surface mappings for generic classes using just category-level image collections (with automatically obtained segmentation). We use images for various categories from ImageNet [**?**], obtain instance segmentation using an off-the-shelf system [**?**], and manually filter out instances with heavy occlusion. This results in about 1000 instances per category, and we train our CSM predictors using a per-category template model downloaded from the web (in fact, for zebras we use a horse model). We show qualitative results (on held-out images) in Figure **??** and observe that we learn accurate mappings that also respect correspondence across instance. Please see supplementary for additional visualizations.

2.5 Discussion

We present an approach to learn canonical surface mappings for generic categories using a geometric cycle consistency objective. Our approach allows us to do so without keypoint or pose supervision, and learn CSM prediction and infer dense correspondence while only relying on foreground masks as supervision. While this is an encouraging step towards understanding the underlying 3D structure and associations across images, several challenges still remain. Though the shape underlying instances in a category may be similar, the instances can often articulate. It would be interesting to extend our approach to also allow for predicting the underlying deformation in addition to camera transforms. Additionally, while our approach allowed relaxing correspondence supervision, it would be desirable to take a step further, and learn from unannotated image collections without foreground mask supervision. Lastly, our approach leveraged geometric cycle consistency, and videos may provide an additional learning signal by enforcing consistency of predictions through time [**?**].

2.6 Results on Internet Videos

In the supplementary video we show results of our method on several videos. The color map on the video sequences shows correspondence to the template shape – shown at the top right of the frame. This helps us understand and visualize intra-frame correspondences. They also show the consistency of our predictions across frames. For instance, similar colors for the tails of two birds indicates that these pixels map to similar points on the template shape. We see few snapshots from the videos in the Fig **??**. It is important to note that since we are using segmentation masks from pre-trained Mask-RCNN, the failure modes of Mask-RCNN become our failure modes. We observe

that false-detections and failure to detect the instance in certain frames results in absence of CSM. Furthermore, since we only train using isolated untruncated and unoccluded objects, our predictions are often inaccurate if objects overlap or are not fully visible.
It is important to note that we do not apply any smoothing or consistency across frames. Our method operates on all the frames in the video independently.

2.7 Additional Result Visualization

We show additional results on all the categories in Figure **??**, **??**, **??**, **??**, **??**, **??**

H Training Details

H.1 Network Archicture

We use a 5 Layer UNet [**?**] with 4 $\times 4$ convolution at each layer. The UNet takes input an image and learns to predict an unit-vector which parameterizes u, v. Along with that we also train the UNet to predict a segmentation of the object which is necessary for keypoint evaluations. We train our networks for over 200 epochs on all the datasets independently. We use Adam [**?**] for optimization of our neural network with a learning rate of 10^{-4}.

H.2 Optimization

Pose Prediction We predict N (=8) possible hypothesis for pose given an image. We initialize the poses such that they span a wide spectrum during start of the training. We add an additional loss to encourage diversity and to ensure there is no mode collapse. The diversity loss consists of two terms:

- We add an entropy term over the probabilities of hypothesis c_i which prevents mode collapse, and encourages exploration. This is equivalent to minimizing $\sum_i^N c_i \log(c_i)$
- We maximize a pair-wise distance between predicted quaternions, $\text{Dist}(q_i, q_j)$ for all the predicted hypothesis of an instance. This is equivalent to minimizing $\sum_{i=1}^N \sum_{j=1, j\neq i}^N \text{Dist}(q_i, q_j)$

I Evaluation Metrics

Keypoint Transfer AP (APK). Any keypoint transfer method given two images as input helps us infer how keypoints transfer from a source image to a target image. The method give two outputs for every keypoint a) transferred keypoint location b) confidence score. A keypoint transfer is successful if the confidence score of the method for the transfer is high and the error for the transfer is less than $d = \alpha \times \max(h, w)$, where h, w represent height and width respectively. For any method we create several confidence thresholds compute the following metrics. Let us consider we have a lot of image-pairs where we have only N_{pair} keypoint correspondences between source and target. For any given confidence threshold t following are the two cases:-

1. True Positive (TP): The confidence for the correspondence was above t, and the transfer error is less than d.
2. False Positive (FP): The confidence for the correspondence was above t, but either the given keypoint does not exist on the target image, or our transfer error is more than d.

We compute transfer precision and transfer recall as follows

$$\text{Transfer Precision} = \frac{N_{TP}}{N_{TP} + N_{FP}}$$
$$\text{Transfer Recall} = \frac{N_{TP}}{N_{\text{pair}}}$$

Here, N_{TP} represents number of True Positives and N_{FP} represents number of False Positives. We create the plots for transfer precision vs transfer recall as shown in the Figure 7 in the main manuscript. Area under such a plot represents AP and we report performance on the same in Table 1 in the main manuscript.

J Results on Internet Videos

In the supplementary video we show results of our method on several videos. The color map on the video sequences shows correspondence to the template shape – shown at the top right of the frame. This helps us understand and visualize intra-frame correspondences. They also show the consistency of our predictions across frames. For instance, similar colors for the tails of two birds indicates that these pixels map to similar points on the template shape. We see few snapshots from the videos in the Fig **??**. It is important to note that since we are using segmentation masks from pre-trained Mask-RCNN, the failure modes of Mask-RCNN become our failure modes. We observe that false-detections and failure to detect the instance in certain frames results in absence of CSM. Furthermore, since we only train using isolated untruncated and unoccluded objects, our predictions are often inaccurate if objects overlap or are not fully visible.
It is important to note that we do not apply any smoothing or consistency across frames. Our method operates on all the frames in the video independently.

K Additional Result Visualization

We show additional results on all the categories in Figure **??**, **??**, **??**, **??**, **??**, **??**

Figure 2.6: Predicted Canonical Surface mapping for six different categories. The color at each image pixel depicts the color at the corresponding surface point on the 3D template shape in the left row. While the predictions are mostly accurate, some error modes include: a) inferring globally incorrect CSM due to pose ambiguity (*e.g.* third horse), or b) incorrect local predictions due to missing segmentation (*e.g.* last cow).

Annotation	Method	Birds		Cars	
		PCK	APK	PCK	APK
KP + Seg. Mask	CMR [**?**]	46.9	20.3	45.4	18.0
Pose + Syn. Data	Zhou et. al [**?**]	-	-	37.1	10.5
Pose + Seg. Mask	CSM (ours) w/ pose	56.6	31.6	53.0	22.4
Seg. Mask	Dense Equi [**?**]	34.8	11.2	31.6	5.7
	VGG Transfer [**?**]	17.2	2.6	11.7	0.7
	CSM (ours)	47.7	23.1	35.5	8.5

Table 2.1: PCK and APK. Percentage of correct keypoints (PCK) and Keypoint Transfer AP (APK) at $\alpha = 0.1$. See Section **??** for metric descriptions. All evaluations are on 10000 image pairs per category. Higher is better.

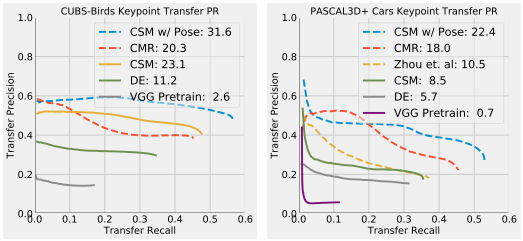

Figure 2.7: Keypoint Transfer PR Curves. We report the transfer precision vs recall curves for all the methods on the task of keypoint transfer. Dahsed lines represent methods with pose or keypoint supervision. Solid lines denote approaches without such supervision. The area under the curve is reported in the legend for each of the plots (higher is better). The plot on left is for CUBS-Birds [**?**], and the one on the right is for cars from Pascal3D+ [**?**]. See Section **??** for metric descriptions.

Figure 2.8: Snaps of a few frames from the Supplementary Video. We downloaded videos from youtube for 6 categories to show our results. We show the template shape in a canonical view on the top-right corner of the image.

Failure Modes Our method has failure modes when the segmentation masks from Mask-RCNN [**?**] are incorrect. Furthmore, since our method is trained on images with a single unoccluded/untruncated object per image hence our predictions are might be in-accurate for occluded objects or partially visible objects

Figure 2.9: Results of randomly sampled birds from the validation set

Figure 2.10: Results of randomly sampled cars from the validation set

Figure 2.11: Results of randomly sampled horses from the validation set

Figure 2.12: Results of randomly sampled zebras from the validation set

Figure 2.13: Results of randomly sampled cows for the validation set

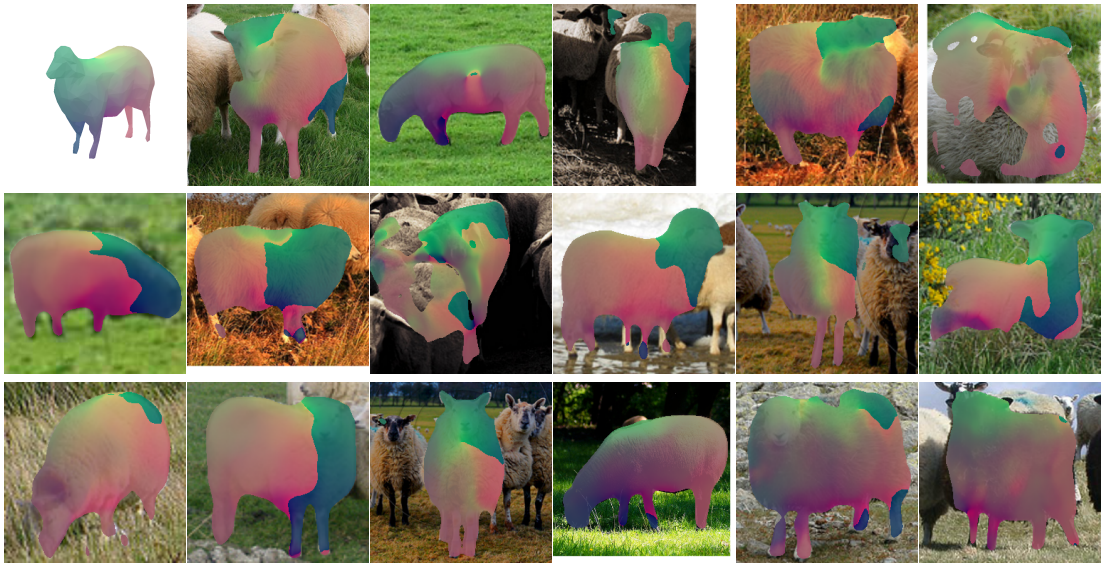

Figure 2.14: Results of randomly sampled sheeps from the validation set

Figure 2.15: Snaps of a few frames from the Supplementary Video. We downloaded videos from youtube for 6 categories to show our results. We show the template shape in a canonical view on the top-right corner of the image.

Failure Modes Our method has failure modes when the segmentation masks from Mask-RCNN [**?**] are incorrect. Furthmore, since our method is trained on images with a single unoccluded/untruncated object per image hence our predictions are might be in-accurate for occluded objects or partially visible objects

Figure 2.16: Results of randomly sampled birds from the validation set

Figure 2.17: Results of randomly sampled cars from the validation set

Figure 2.18: Results of randomly sampled horses from the validation set

Figure 2.19: Results of randomly sampled zebras from the validation set

Figure 2.20: Results of randomly sampled cows for the validation set

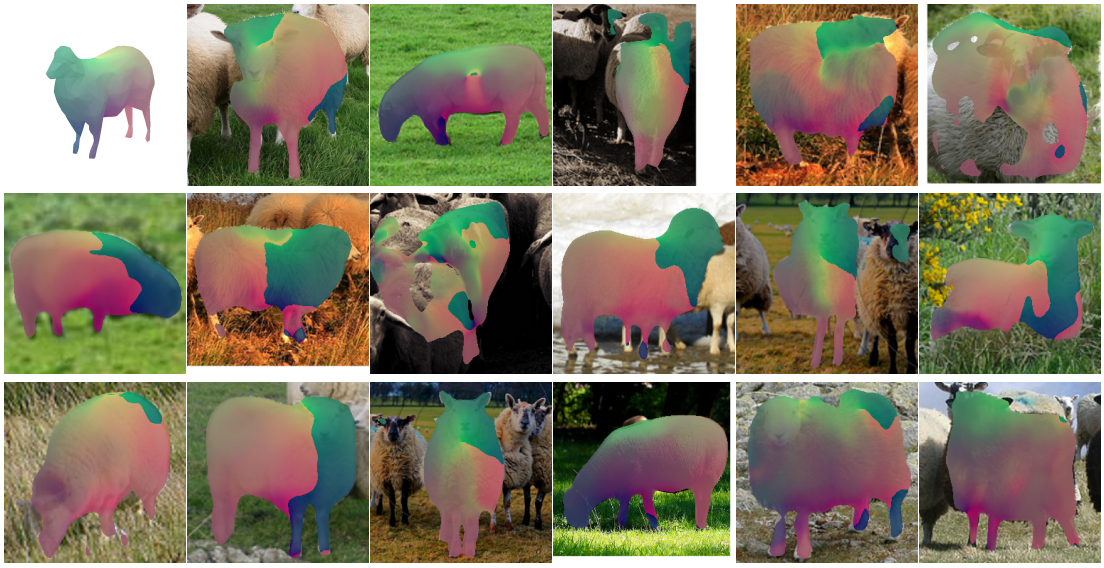

Figure 2.21: Results of randomly sampled sheeps from the validation set

Chapter 3

3D-RelNet: Joint Object and Relational Network for 3D Prediction

(a) Overview of our method (b) Results

Figure 3.1: (a) Approach Overview: We study the problem of layout estimation in 3D by reasoning about relationships between objects. Given an image and object detection boxes, we first predict the 3D pose (translation, rotation, scale) of each object and the relative pose between each pair of objects. We combine these predictions and ensure consistent relationships between objects to predict the final 3D pose of each object. **(b) Output:** An example result of our method that takes as input the 2D image and generates the 3D layout.

1 Introduction

A single 2D image can induce a rich 3D perception. When we look at an image, we can reason about its 3D layout, the objects in the image, their shape, extent, relationships *etc*. This is really surprising given that going from a 2D projection to a 3D model is inherently ill-posed. How are we able to solve this problem? Humans rely on regularities in the 3D world in order to do so – this helps us discard many improbable solutions in 3D and reason about more likely ones. This regularity exists at the scene level - indoor scenes have roughly perpendicular walls; object level - chairs have similar shapes; and in local object relationships - chairs are close to tables, monitors are on top of tables *etc*. A decade ago, a lot of work in computer vision focused on using all the three levels of regularities. For example, a lot of work focused on object-centered 3D prediction [**?**], scene-level 3D prediction [**?**], and multi-object 3D reasoning [**?**]. However, in recent years, since the advent of ConvNets, a vast majority of computer vision approaches do not leverage these object-object relationships, and instead reason about each object independently.

In this paper, we attempt to take a holistic view of the 3D prediction problem and note that solving

the 3D prediction problem would require incorporation of all the three cues. We believe there are three fundamental questions that need to be answered to design this holistic architecture: (a) What is the right representation for object level 3D prediction?; (b) How do we represent object-object relationships and how do we predict them from pixels?; (c) Finally, how to incorporate object-object relationships with object-level modules. This paper builds upon the recent success in (a) and investigates how to model relationships and incorporate them into our 3D prediction framework.

So, how do we model relationships and estimate them from pixels? There is a whole spectrum of possible approaches. On one end of the spectrum is a complete end-to-end approach. Some examples of these include Interaction Networks [**?**] or Graph Convolutional Networks [**?**]. Both these methods provide a mechanism for object features in the scene to effect each other, thereby allowing an implicit modeling of relationships among them. However, as we show in our experiments, these end-to-end approaches disregard the structural information which might be crucial for modeling the relationships. The other end of spectrum is to use category-based image-agnostic pairwise priors [**?**] to model relationships. A drawback is that these priors are often too strong to generalize and it is better to learn them [**?**].

When it comes to the final question of how does one incorporate relationships to improve 3D prediction, the answer is even murkier. One classical approach is to use graphical models such as CRFs [**?**, **?**]. However, these classical approaches have usually provided little improvements over object-based approaches. Our key insight is to incorporate structural information in end-to-end systems. Specifically, we model and predict pairwise relationships in the translation, rotation and scale space. One advantage of using this structured relationship space is that the incorporation of relationships into object-level estimates is simple yet effective. But how do we predict these pairwise relationships from pixels? Our paper investigates several design choices and proposes a simple architecture. Our method demonstrates significant improvement in performance across multiple metrics and datasets. As we show in our experiments, this modeling of relationships in this structured space provides a huge **6 point AP** improvement in detection settings over current state-of-the-art 3D approaches. We will release our code for reproducibility.

2 Related Work

Dating back to the first thesis written on computer vision [**?**], inferring the underlying 3D structure of an image has been a long-standing goal in the field. While we have explored several 3D scene representations over the years, *e.g.*, depth [**?**, **?**], qualitative 3D [**?**, **?**], manifolds [**?**] or volumetric 3D [**?**, **?**, **?**] , the prevalent paradigm is still the one followed in Roberts' seminal work – that of inferring a 3D scene in terms of the shape and pose of the underlying objects.

The initial attempts under this paradigm [**?**, **?**] focused on placing known object instances to match image evidence, relying on matching edges, corners *etc*. to fit the known shape templates to images. Subsequent approaches have focused on a more general setting of reconstructing scenes comprising of novel objects, and leverage either explicit or implicitly learned category level priors for pose and shape estimation, typically relying on a deformable shape space [**?**, **?**] or template CAD models [**?**, **?**, **?**, **?**, **?**] for the latter inference. Current CNN based incarnations of these approaches, driven by the abundant success of deep learning and availability of annotated data, have further improved the results for pose estimation [**?**, **?**], and have also been extended to joint shape and pose inference of the objects present in a scene [**?**, **?**].

A common characteristic amongst these approaches is the reasoning at a per-object level. While the object-centric nature is certainly desirable as a representation, we argue that reasoning independently for each object to infer this representation is not, as it does not allow leveraging the relationships

between the entities in a scene. We propose a method that also uses object-centric representations but goes beyond independent reasoning per object.

We are of course not the first to pursue reasoning about relationships between entities in a scene. Several previous approaches focus on the goal of predicting various relations e.g. human-object interactions [**?**, **?**], object-object interactions [**?**, **?**], object-scene interactions [**?**] *etc*. While these works pursue relation inference as the end goal, we instead aim to leverage these for a per-instance prediction task. In the context of incorporating relations for such per-instance prediction, there are two alternate ideologies. On the one hand, approaches pursuing 3D scene inference or generation [**?**, **?**, **?**, **?**, **?**, **?**, **?**] typically incorporate pairwise (or higher order) relations via explicit class-based priors regarding possible configurations and optimize predictions to adhere to these. This approach of explicitly modeling relations as a prior imposes the same constraints across all scenes, independent of the structure in the image, and is therefore not flexible enough and has difficulties in scaling up to arbitrary relations across arbitrary objects.

The alternate ideology for incorporating relationships is to eschew any explicit structure for these relations, and instead implicitly capture these via architectural changes to the CNNs, thereby allowing the features of objects to influence one another [**?**, **?**]. While this a generally applicable mechanism, it does not leverage several aspects regarding the structure of the problem – for 3D inference, specific relations like relative position, orientation are very relevant, and can be used in specific ways to influence per-instance predictions. Our approach leverages some aspects of both these ideologies – unlike the classical prior based approaches, we learn and infer these relations in a image-dependent context via a CNN, and unlike the purely implicit methods, we are more explicit about the structure and meaning of these relations.

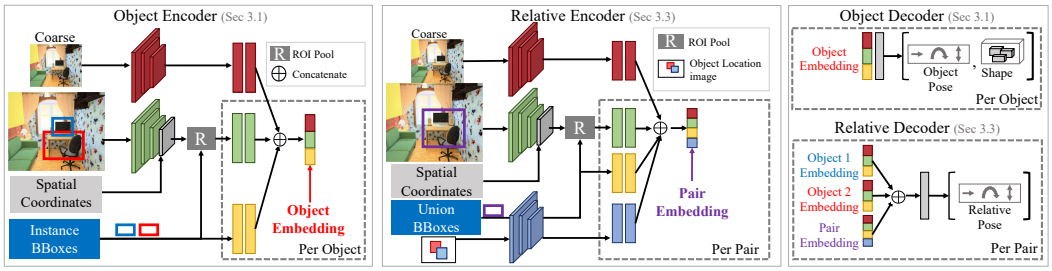

Figure 3.2: Approach Details: We use the instance encoder to create an embedding for each object (instance) in the scene. The instance decoder uses this embedding to predict a pose for each object independently. The relative encoder takes each pair of instances and their embeddings to output an embedding for the pair. The relative decoder predicts a relationship (relative pose) between the pairs. We combine these relative and per-object predictions to predict final pose estimates for each object in an end-to-end differentiable framework.

3 Approach

Our goal is to predict the 3D pose and shape for all the objects in a scene. We observe that in addition to the visual cues per object, reasoning about relationships across them can further help our predictions, in particular for the 3D pose – a chair would be in front of a table, and of a compatible relative size, and therefore even if we are uncertain about the pose of one of these objects, *e.g.*, due to occlusion, these relationships can enable us to make accurate predictions.

We operationalize this insight in our method (see figure **??**) that leverages both - independent per-object predictions alongwith predictions regarding relationships between them. We infer the final estimates for all the objects in the scene by integrating these two. We first formally describe the

object-centric representations pursued and briefly review a recent per-instance prediction approach in section **??**. We then introduce the relative representations in section **??** and present our network architecture that enables predicting these in section **??**. In section **??** we discuss how these relative predictions are combined with the independent per-object predictions to yield the final 3D estimates for the objects. We show that optimizing the combination of these estimates (*i.e.*, final estimate) in a differentiable end-to-end framework helps improve the final per-object predictions.

3.1 Instance Specific Representations and Inference

We output the 3D pose of an object by predicting its shape in a canonical frame, and its scale, translation and rotation in camera frame. The shape is parametrized as a 32^3 volumetric occupancy grid in a canonical space where the objects are upright, front-facing, normalized to a unit cube. The translation $\mathbf{t} \in \mathbb{R}^3$ and (logarithm of) anisotropic scale $\mathbf{s} \in \mathbb{R}^3$, and normalized quaternion $\mathbf{q}$ indicate the position, size and orientation of the object respectively. Following prior work [**?**], we parametrize the rotation prediction as a classification task among fixed bins.

We build upon the recent work by [**?**]: they pursue a similar per-object representation, but make independent predictions across objects. We use their approach to obtain the independent predictions for each object. We briefly review their prediction framework but refer the reader to the paper for details about the representation and architecture. Their method uses an architecture similar to Fast R-CNN [**?**], where the input image is encoded via convolutional layers, and for each object bounding box, an RoI pooling layer crops the corresponding features. These per-object features, in conjunction with coarse image feature and an encoding of the bounding box coordinates, are encoded to an instance-specific bottleneck representation from which the corresponding shape and pose are predicted.

We adopt a similar architecture, illustrated in Figure **??**, with the introduction of spatial coordinates as additional feature channels (see Section **??**) to obtain per-object (unary) predictions. We note that these predictions are obtained independently across objects, and do not incorporate reasoning across them. However, unlike previous work, these are subsequently integrated with relative predictions (Section **??**) to obtain final estimates.

3.2 Relations as Relative Representations

Given an image of a scene, we can infer that two chairs nearby might be of a similar size, a laptop kept above a desk, and a television facing the bed *etc*. Thus, relative pose between objects captures an important aspect of their relationship in the scene.

Concretely, given two objects, say A and B, we infer the relative pose from A to B. The relative pose, akin to the absolute 3D pose, is factored into the relative translation, scale, and direction. The relative translation is defined as the difference of the absolute locations of the object in the camera frame $\mathbf{t}_{AB} = \mathbf{t}_B - \mathbf{t}_A$. Similarly, the relative scale is simply the ratio of the two object sizes, or equivalently a difference in logarithms $\mathbf{s}_{AB} = \mathbf{s}_B - \mathbf{s}_A$. Finally, we predict a relative *direction* $\hat{\mathbf{d}}_{AB}$, *i.e.*, the direction of object B in the frame of object A: $\hat{\mathbf{d}}_{AB} \propto (\bar{R}_A)\, \mathbf{t}_{AB}$, where $\hat{\mathbf{d}}_{AB}$ is normalized to unit norm. Here $\bar{R}_A$ denotes the rotation corresponding to q_A. We note that this parametrization, unlike relative rotation, helps us overcome some ambiguities due to symmetries, *e.g.*, the relative direction from a chair to a table in front of it is unambiguous, even if the table is symmetric.

3.3 Network Architecture

Our network has two branches – a per-object (unary) prediction module, and a relative prediction module. The architecture of both these branches is shown in figure **??**. The former, as described in section **??**, simply computes a per-object encoding and subsequently makes per-object predictions. The relative prediction module is then tasked with inferring the relative pose (section **??**) for every pair of objects in the scene.

As depicted in figure **??**, the relative prediction module takes as input: a) the encoding of both objects, and b) an encoding of the larger (union) bounding box containing both objects. Since this larger box may contain several additional objects beyond the ones of interest, we additionally give as input two channels with binary masks indicating the source and target bounding box extents respectively (denoted as 'object location image'). We also find it beneficial to concatenate the normalized per-pixel spatial coordinates as additional features, as it allows the network to easily reason about the absolute spatial location of the bounding box(es) under consideration. Finally, similar to the parametrization in section **??**, we frame the relative direction prediction as a classification task among 24 bins, where the bins are computed by clustering relative directions across the training instances like in [**?**].

3.4 Combining Per-object and Relative Predictions

We saw in section **??** that we can obtain independent per-object predictions for the 3D shape and pose, and introduced the relative pose predictions and architecture in section **??** and section **??**. We denote by $(\mathbf{t}_n, \mathbf{s}_n, \mathbf{q}_n)$ the per-object (unary) predictions for the translation, log-scale, and rotation respectively for the n^{th} object, and similarly $(\mathbf{t}_{mn}, \mathbf{s}_{mn}, \mathbf{d}_{mn})$ denote the relative pose predictions from the m^{th} to the n^{th} object. Using these, we show that we can obtain final per-instance predictions $(\mathbf{t}^*_n, \mathbf{s}^*_n, \mathbf{q}^*_n)$ that incorporate both, the unary and relative predictions.

Translation and Scale Prediction. The relative predictions give us linear constraints that the final predictions should ideally satisfy. As an example, we would want $\mathbf{t}^*_n - \mathbf{t}^*_m = \mathbf{t}_{mn}$. This can equivalently be expressed as $A_{mn}\mathbf{t}^* = \mathbf{t}_{mn}$, where A_{mn} is a sparse row-matrix with the m^{th} and n^{th} entries as $(-1, 1)$, and $\mathbf{t}^*$ denoting the final translations for all the N objects. We can similarly express all pairwise linear constraints as $A\mathbf{t}^* = \mathbf{t}_{\text{rel}}$, where $\mathbf{t}_{\text{rel}}$ denotes all the relative predictions, and A is the appropriate sparse matrix.

In addition to satisfying these linear constraints, we would also like the final estimates to be close to the unary predictions. We can therefore incorporate both, the relative and the unary constraints via a system of linear equations, and solve these to obtain the final estimates.

$$\begin{bmatrix} \lambda\, I \\ A \end{bmatrix} \mathbf{t}^* = \begin{bmatrix} \lambda\, \mathbf{t} \\ \mathbf{t}_{\text{rel}} \end{bmatrix}; \quad \mathbf{t}^* = \begin{bmatrix} \lambda\, I \\ A \end{bmatrix}^{+} \begin{bmatrix} \lambda\, \mathbf{t} \\ \mathbf{t}_{\text{rel}} \end{bmatrix} \tag{3.1}$$

Here X^+ denotes the Moore-Penrose inverse of matrix X, and λ indicates the relative importance of the unary estimates. We can therefore obtain the final translation predictions $\mathbf{t}^*$ that integrate both, unary and relative estimates. We note that the final estimates are simply a linear function of the unary predictions $\mathbf{t}$ and the relative predictions $\mathbf{t}_{\text{rel}}$, and it is therefore straightforward to propagate learning signal from supervision for $\mathbf{t}^*$ to these predictions. While the description here focused predicting translation, as we represent scale using logarithm of the sizes, similar linear constraints apply. We can therefore similarly compute final predictions for the scales across objects $\mathbf{s}^*$.

Rotation Prediction. While the incorporation of unary and relative predictions can be expressed via linear constraints in the case of translation and scale, a similar closed form update does not apply for

rotation because of the framing as a classification task and non-linearity of the manifold. Instead, we update the likelihood of the unary predictions based on how consistent each rotation bin is with the relative estimates.

We denote as R^b the rotation matrix corresponding to the b^{th} rotation bin, and use $\Delta(R, \mathbf{d}, \mathbf{t})$ to measure how inconsistent a predicted rotation R is w.r.t the predicted relative direction distribution $\mathbf{d}$ and relative translation $\mathbf{t}$ (see appendix for details). Using these, we can compute the (unnormalized) negative-log likelihood distribution over possible rotations as follows:

$$\mathbf{q}^*_m(b) = \mathbf{q}_m(b) + \sum_n \Delta(R^b, \mathbf{d}_{mn}, \mathbf{t}_{mn}) \tag{3.2}$$

This update to compute $\mathbf{q}^*$ can equivalently be viewed as a single round of message passing, with the messages being of an explicit rather than an implicit form.

Training Details. We described above how the independent per-object predictions are analytically combined with the relative pose predictions to obtain the final estimates, and note that this integration process allows us to propagate learning signal from supervision on the final estimates back to the unary and relative predictions. Our training happens in two steps. In the first step, we train both unary and relative predictions independent of each other. We formulate the loss-function for each network similar to [**?**]. Specifically, we use regression losses for shape encoding, the (absolute and relative) translation and scale, and classification losses for the rotation and relative direction prediction. Note that as some objects might be rotationally symmetric, we allow multiple 'correct' bins for these and maximize the maximum probability across these. In the second step, after a few epochs, we train the whole model in joint manner. We add similar losses for the final pose predictions that are computed using both, unary and relative estimates. During inference, given the unary and relative predictions, we simply compute the final pose predictions via the optimization process described above. Additional details on optimization are provided in the appendix.

4 Experiments

4.1 Experimental Setup

Datasets: We use the SUNCG dataset [**?**] which provides many diverse and detailed 3D models for houses. Following [**?**, **?**], we use the 2D renderings of the houses and the corresponding parsed 3D house model information to get roughly 550k image and 3D supervision pairs. We follow the setup in [**?**] and split this data into train (70%), val (10%) and test (20%); with objects classes - bed, chair, desk, sofa, table, and tv.

We also use the NYUv2 [**?**] dataset which consists of 1449 real world images, and use the annotations by [**?**] to finetune the network trained on SUNCG using the same subset of object categories. This dataset has lower resolution images and serves well to check the generalization of our approach to real data. As the NYUv2 annotations use the same small set of 3D shapes across train and test images, we do not evaluate shape prediction.

During inference, the input to our system is always a 2D image along with 2D bounding boxes indicating the objects in the scene. We present results in both scenarios: a) when the 2D boxes correspond to ground-truth locations, and b) when the 2D boxes are obtained via a detector. Given this input, our method outputs a 3D prediction for each of the boxes. We summarize this in Table **??** and provide pointers to the results under the two settings.

Metrics: [**?**] propose different metrics to measure the quality of various prediction components - detection, rotation, translation and scale. They also propose different thresholds δ for each of these

Section	Input	Output
Sec **??**	2D Images, ground truth boxes Table **??**, Fig **??**, **??**	3D Pose for each box
Sec **??**	2D Images, detection boxes Table **??** Fig **??**	3D Pose for each box
Sec **??**	2D Images, ground truth boxes Tables **??**, **??**	3D Pose for each box

Table 3.1: We provide an overview of the **test time** experimental settings we use in Section **??**. We define the datasets, inputs and outputs.

components which are used to count a prediction as a true positive in the detection setting. We use these metrics and refer the reader to the appendix for a summary review.

Baselines: We use the following baseline methods:

- *Factored3D*: This method from [**?**] reasons about each object independently and serves as a baseline to see how our relationship reasoning can improve performance.
- *Factored3D + CRF*: We optimize the independent predictions from [**?**] using a CRF [**?**] with unary and binary terms. The binary potential terms correspond to log-likelihood of the relative pose for each pair, where the likelihood is modelled using a mixture of gaussian model for each category pair. Note that this baseline is allowed to use the ground-truth object classes during inference. See appendix for details.
- *GCN*: We use Graph Convolutional Networks [**?**] to perform implicit relational reasoning. We use the Object Encoder (Figure **??**) to obtain object embeddings o_i for each object and then use a 2-layer GCN to obtain the final object embeddings for each object. These are then used to predict the 3D pose and shape for the objects.
- *InteractionNet*: We use the method from [**?**] as an alternate way to perform implicit reasoning over the object embeddings. We compute an 'effect' embedding e_{AB} for each ordered tuple (o_A, o_B) via a learned MLP, and update the object embeddings by aggregating these as $o_A + \max_B(e_{AB})$, and use these for per-object predictions.

Note that while the latter two baselines can implicitly reason about relationships, they ignore the underlying structure (how relative translations affect translation *etc*.). In contrast, while the CRF baseline leverages this structure, it ignores the image when reasoning about relationships – the relative pose prior is image agnostic, whereas in our approach the relative pose is predicted.

4.2 Evaluation Using Ground Truth Boxes

We first analyze all methods in the setting where we are given the ground truth bounding boxes. In this setting, we can analyze just the 3D prediction quality without the additional variance introduced due to imperfect detection. During training, we train all the methods on ground-truth boxes as well as object proposals (obtained using [**?**]) which have an IOU ≥ 0.7. Each method is trained to predict the 3D pose of the objects.

At test time, we evaluate all the methods by providing the image and the ground truth boxes as input. All methods predict the translation, rotation and the scale for each of the ground truth boxes. As shown in Table **??**, our method generates higher quality predictions across both translation and scale where it can outperform the baselines on the mean, median and % error. First we observe the

Figure 3.3: 3D Output using Ground truth 2D Boxes: In Section **??**, we evaluate only the 3D output of all methods by using ground truth boxes at test time. We visualize the 3D predictions for our method and the baseline. Our method estimates 3D better than the baseline, *e.g.*, bottom right where the distance between the chairs and table is predicted correctly, and the pose of the yellow chair is corrected. Best viewed in color.

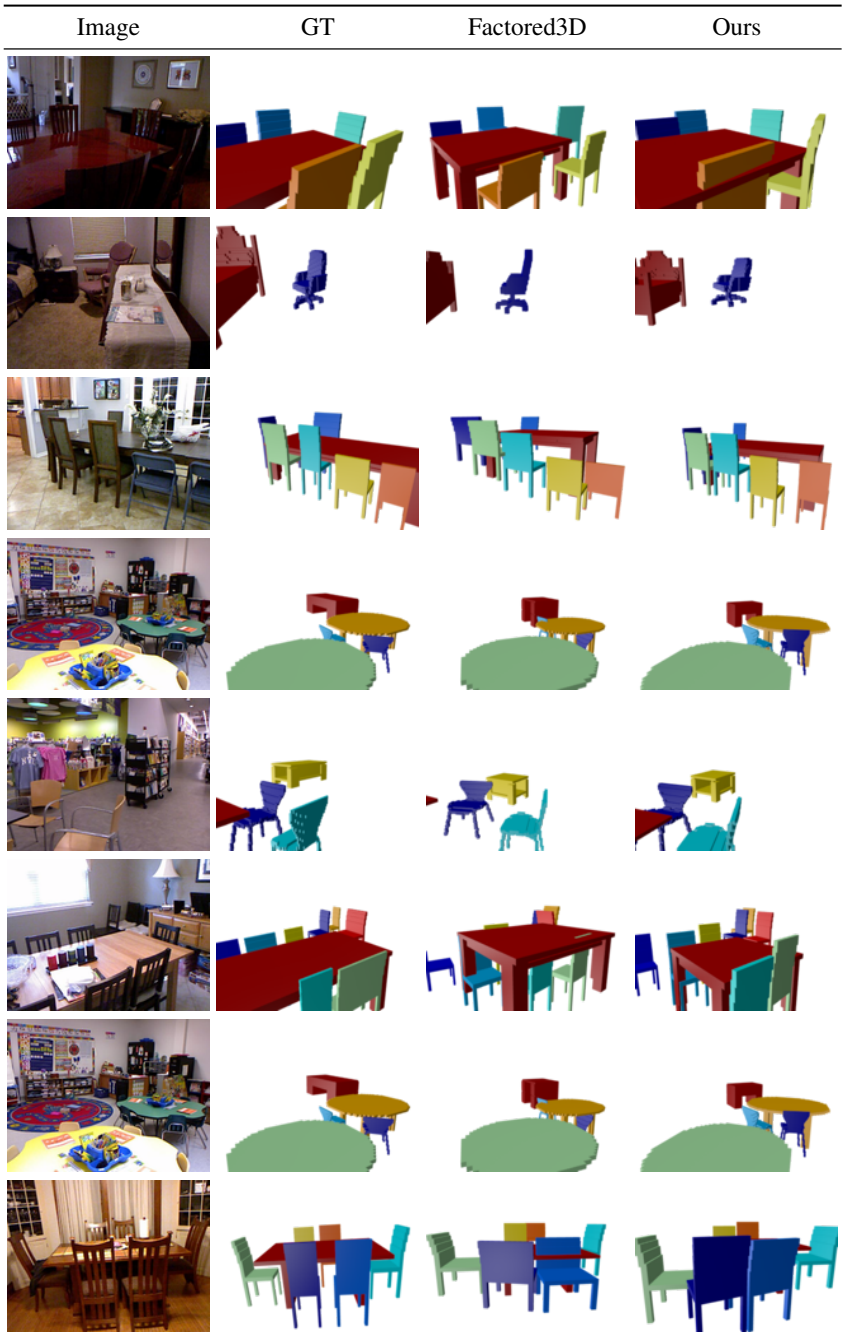

Figure 3.4: Ground truth boxes for NYU: We visualize sample prediction results on the NYUv2 dataset in the setting with known ground-truth boxes. Our method produces better estimates than the baseline, *e.g.*, for the top left image we see that the chairs are better aligned by our method as compared to the baseline. We use the ground-truth meshes for visualization due to lack of variability in shape annotations for learning.

performance of the CRF [**?**] model and conclude that there is not a significant gain over the baseline and minor improvements to translation. It is also worth noting that adding any form of relationships modeling, like in GCN or InteractionNet, gives a boost over doing per-object predictions like in Factored3D. Our structured reasoning and inference about object relationships provides a boost over other pairwise models such as GCN and InteractionNet. Our performance gain over the baseline holds across SUNCG and NYUv2 datasets demonstrating the generalization of our method.

Qualitative Results: In figure **??**, we show a few results of our method and the baseline (Factored3D). We see that our method can correct many error modes (relative positions and poses of objects) compared to the baseline. We observe similar trends on the NYUv2 dataset (figure **??**).

Dataset	Method	Translation (meters)			Rotation (degrees)			Scale		
		Median (lower is better)	Mean	(Err ≤ 0.5m)% (higher is better)	Median (lower is better)	Mean	(Err $\leq 30^\circ$)% (higher is better)	Median (lower is better)	Mean	(Err ≤ 0.2)% (higher is better)
SUNCG	Factored3D	0.28	0.39	79.5	**4.56**	19.91	86.4	0.16	0.25	58.4
	CRF	0.27	0.38	79.7	4.59	20.18	86.1	0.16	0.26	57.4
	InteractionNet	0.28	0.37	80.0	4.58	20.19	86.4	0.11	**0.19**	68.6
	GCN	0.26	0.38	79.3	4.60	20.45	86.0	0.11	0.20	69.1
	Ours	**0.23**	**0.33**	**84.0**	4.58	**19.82**	**86.6**	**0.10**	**0.19**	**69.7**
NYUv2	Factored3D	0.49	0.62	51.0	14.55	42.55	63.8	0.37	0.40	18.9
	Interaction Net	0.45	0.59	56.2	**13.34**	**38.7**	**67.6**	0.36	0.39	20.1
	GCN	0.45	0.60	55.6	14.22	41.63	65.7	0.37	0.40	19.2
	Ours	**0.41**	**0.54**	**60.9**	14.00	39.60	67.0	**0.33**	**0.38**	**21.7**

Table 3.2: We use **ground truth boxes** at test time and compare the 3D prediction performance of the methods (see Section **??**). We show the median and mean error (lower is better) and the % of samples within a threshold (higher is better).

4.3 Evaluation Using Detections

In this setting, we test the learned models using detections from a pre-trained object detector (taken from [**?**]). Thus, we can now evaluate the robustness of these methods when the input object boxes are not pristine. In Table **??**, we see the mean Average Precision values for different criteria used to define true positives (see appendix for metrics). As an example, in the first column a true positive satisfies IOU (box2d) threshold with the ground truth, is close in scale, rotation, translation, shape thresholds. Each subsequent column examines one criteria so we can analyze their accuracy with detections. We see (in the first column) that our method provides a significant gain of **6 points** in mAP over the baseline. Relationship modeling (GCN and InteractionNet) provides benefit in the detection setting too. However, our structured reasoning outperforms these methods, with most of the gains coming from predicting higher quality scale and translation. We visualize some predictions showing the difference between our approach and the baseline in figure **??**. Furthermore, in Table **??** we evaluate the our method on the NYUv2 in the detection setting, and we achieve similar trends in performance with respect to the baseline. Due to increased difficulty of the task on NYU we observe relative lower mAP scores.

4.4 Effect of Combination and Optimization

Combining the unary and relative predictions provides benefit at the training time because it allows the model to modify its unary and relative predictions so that they are more 'compatible' with each other. We quantify this in Table **??** using the SUNCG dataset. We report the performance of a purely multi-task version ('MT') that only predicts the unaries and relative pose values, without ever combining them. The 'MT + combine test only' combines these during inference (but not

Figure 3.5: Detection Setting for SUNCG: We visualize sample prediction results in the detection setting (Section **??**). Our method produces better estimates than the baseline, *e.g.*, in the first image the television set and the table are better placed. The colors only indicate separate instances, and do not correspond between the ground-truth and the predicted representations. We show 3D outputs (detection setting) for NYU in the supplementary material

Method	Translation (meters)			Rotation (degrees)			Scale		
	Median (lower is better)	Mean	(Err $\leq$ 0.5m)% (higher is better)	Median (lower is better)	Mean	(Err $\leq 30^\circ$)% (higher is better)	Median (lower is better)	Mean	%(Err $\leq$ 0.2)% (higher is better)
Multi-task (MT)	0.25	0.35	81.9	4.55	19.33	86.7	0.11	0.20	68.2
MT + combine test only	**0.23**	0.34	83.5	**4.51**	**19.30**	**86.9**	0.11	**0.19**	68.4
MT + combine train only	0.25	0.36	82.0	4.63	20.02	86.1	0.11	**0.19**	68.9
MT + combine train & test (Ours)	**0.23**	**0.33**	**84.0**	4.58	19.82	86.6	**0.10**	**0.19**	**69.7**

Table 3.3: We study the effect of **combining the unary and relative predictions** at various stages. The multi-task model does not combine them, whereas the next two models combine them either at train or test time. Our method that jointly optimizes these predictions (and their combination) at both train and test time shows the biggest improvement.

Method	Translation (meters)			Rotation (degrees)			Scale		
	Median (lower is better)	Mean	(Err $\leq$ 0.5m)% (higher is better)	Median (lower is better)	Mean	(Err $\leq 30^\circ$)% (higher is better)	Median (lower is better)	Mean	%(Err $\leq$ 0.2)% (higher is better)
Ours $-$ spatial	0.24	0.34	83.4	4.77	18.81	82.4	**0.10**	**0.19**	**70.4**
Ours $-$ mask	0.24	0.35	82.7	4.47	19.52	86.1	0.17	0.26	56.1
Ours $-$ spatial $-$ mask	0.25	0.35	82.7	**4.47**	**19.18**	**86.9**	0.15	0.23	61.8
Ours	**0.23**	**0.33**	**84.0**	4.58	19.82	86.6	**0.10**	**0.19**	69.7

Table 3.4: We analyze the benefit of adding **spatial coordinate** features to all the methods, and using the **object location mask** (section **??**) in our method. We remove these features and measure performance on SUNCG. We report the median and mean error (lower is better) and the % of samples within a threshold (higher is better).

training). Finally, the 'MT + combine train only', where we also compare to a method that only uses this combination at train time, and finally report our full method as a reference. We see that combining the relative predictions at either train or test alone performs better than pure multi-task learning. This shows the importance of the unary and relative predictions interacting with each other. Combining these at both train and test time, and jointly optimizing like in our method performs the best.

Our experiments in section **??** demonstrate the qualitative and quantitative advantages of modeling relationships for 3D estimation using our method. In this section we use the SUNCG dataset to analyze some architecture design choices.

Effect of Spatial Coordinates. Our method appends spatial coordinates (following [**?**]) to improve the final prediction. In Table **??**, we study the effect of adding these spatial coordinates on our method and the baselines. We show the results of our method without spatial coordinates in row 1.

Effect of Object Location Masks. Our method also appends the mask of the object pair to the input to the relative prediction network, and in Table **??** we analyze the effect of removing mask. Comparing the results of our method (row 4) to when we remove the location masks (row 2), we note that location masks improve the translation and scale predictions.

5 Discussion And Future Work

We proposed a method to incorporate relationship based reasoning in the form to relative pose estimates for the task of 3D scene inference. While this allowed us to significantly improve over existing approaches that reason independently across objects, numerous challenges still remain to be addressed. In our approach, we only leveraged pairwise relations among the objects in a scene,

Dataset	Method	all	box2d + trans	box2d + rot	box2d + scale
SUNCG	Factored3D	21.72	49.28	**62.77**	33.56
	Interaction Net	26.42	50.26	61.75	41.66
	GCN	24.76	48.63	61.61	39.97
	Ours	**27.76**	**54.38**	62.72	**41.83**
NYUv2	Factored3D	5.30	17.17	20.36	28.36
	Interaction Net	7.57	19.92	20.39	30.93
	GCN	6.49	17.39	21.85	30.42
	Ours	**8.49**	**21.16**	**22.81**	**31.91**

Table 3.5: We report the mean Average Precision (mAP) values for the **detection setting** for SUNCG and NYUv2. In each column, we vary the criteria used to determine a true positive. This helps us analyze the relative contribution of each component (translation, rotation, scale) to the final performance. Higher is better. Note that the scale threshold for NYUv2 is different from the one used on SUNCG

and it would be interesting to pursue incorporating higher order relations. We also relied on a synthetic dataset with full 3D supervision to train our prediction networks, thereby limiting direct applicability to datasets without 3D supervision. Towards overcoming this, it might be desirable to combine our approach with parallel efforts in the community to use 2D reprojection losses [**?**] or leverage domain adaptation techniques [**?**].

F Appendix

F.1 Metrics

We use the metrics from [**?**] and summarize them below.

- Translation ($\mathbf{t}$): Euclidean distance between prediction and ground-truth $\|\mathbf{t}_{\mathrm{p}} - \mathbf{t}_{\mathrm{gt}}\|$. $\delta_{\mathbf{t}} \leq 0.5\mathrm{m}$.
- Scale ($\mathbf{s}$): We measure the average unsigned difference in log-scale, *i.e.*, $\Delta(\mathbf{s}_{\mathrm{p}}, \mathbf{s}_{\mathrm{gt}}) = \frac{1}{3}\sum_{i=1}^{3} |\log_2(\mathbf{s}^i_{\mathrm{p}}) - \log_2(\mathbf{s}^i_{\mathrm{gt}})|$. We threshold at $\delta_{\mathbf{s}} \leq 0.2$.
- Rotation ($\mathbf{q}$): Geodesic distance between rotations $\frac{1}{\sqrt{2}}\|\log(R_{\mathrm{p}}^{\intercal} R_{\mathrm{gt}})\|$. $\delta_{\mathbf{q}} \leq 30^{\circ}$. For objects that exhibit rotational symmetry, we use the lowest error across the different possible values of R_{gt}.
- Shape ($\mathbf{V}$): Following [**?**], we measure the intersection over union (IoU) and use threshold $\delta_V = 0.25$. As a higher IOU is better, so we use $\delta_V \geq 0.25$ for true positive.
- Bounding Box overlap ($\mathbf{b}$): The bounding box overlap is measured using IOU. $\delta_b \geq 0.5$.
- Detection: A prediction is considered a true positive when it satisfies the thresholds for each of the above components $(\delta_{\mathbf{t}}, \delta_{\mathbf{s}}, \delta_{\mathbf{q}}, \delta_V, \delta_{\mathbf{b}})$. We use Average Precision (AP) to measure the final detection performance.

F.2 Training Details

Optimization. We train our network in two stages. In the first stage of training we use ground truth boxes. We train for 8 epochs by using adam optimizer with a learning rate of 10^{-4}. During the first 4 epochs of the training we train for relative and object specific predictions independently and during next 4 epochs of the training we optimize the whole model jointly by combining the relative and object specific estimates. In the next stage we consider overlapping proposals with IOU of over 0.7 with respect to ground truth boxes and the ground truth boxes as positive proposals to further make the model robust in the detection setting.
In the NYUv2 setting we start with a network trained on the SUNCG dataset and finetune the network for 16 epochs on the NYU train + val split and evaluate method on the test split.

Rotation Prediction. We defined $\Delta(R, d, t)$ as a measure of how inconsistent a predicted rotation R is w.r.t the predicted relative direction distribution d and relative translation t. Given a predicted rotation R, we would expect the predicted direction to align with the vector $\bar{R}\,\hat{t}$, where $\hat{t}$ is unit-normalized. Note that the predicted d is a probability distribution over possible directions, and let d^* denote the bin that aligns maximally with $\bar{R}\,\hat{t}$. We measure $\Delta(R, d, t)$ by combining measures of how likely this bin is with how well it agrees with the rotation and translation: $\Delta(R, d, t) = -\log\, p(d^*) + (1 - \cos(d^*, \bar{R}\,\hat{t}\,))$.

Relative Importance. We use lambda for unary importance to get $\mathbf{t}^*$ and $\mathbf{s}^*$ as 1. In case of rotation we use we weight for the relative predictions as $\min(5.0/n, 1)$ where n represents number of neighbours of the object. In the detection setting we create set of valid objects which are allowed to influence the final predictions for other objects based upon the detection score. We consider objects with a score above 0.3 to be part of the valid set and only use them to get final predictions for other objects.

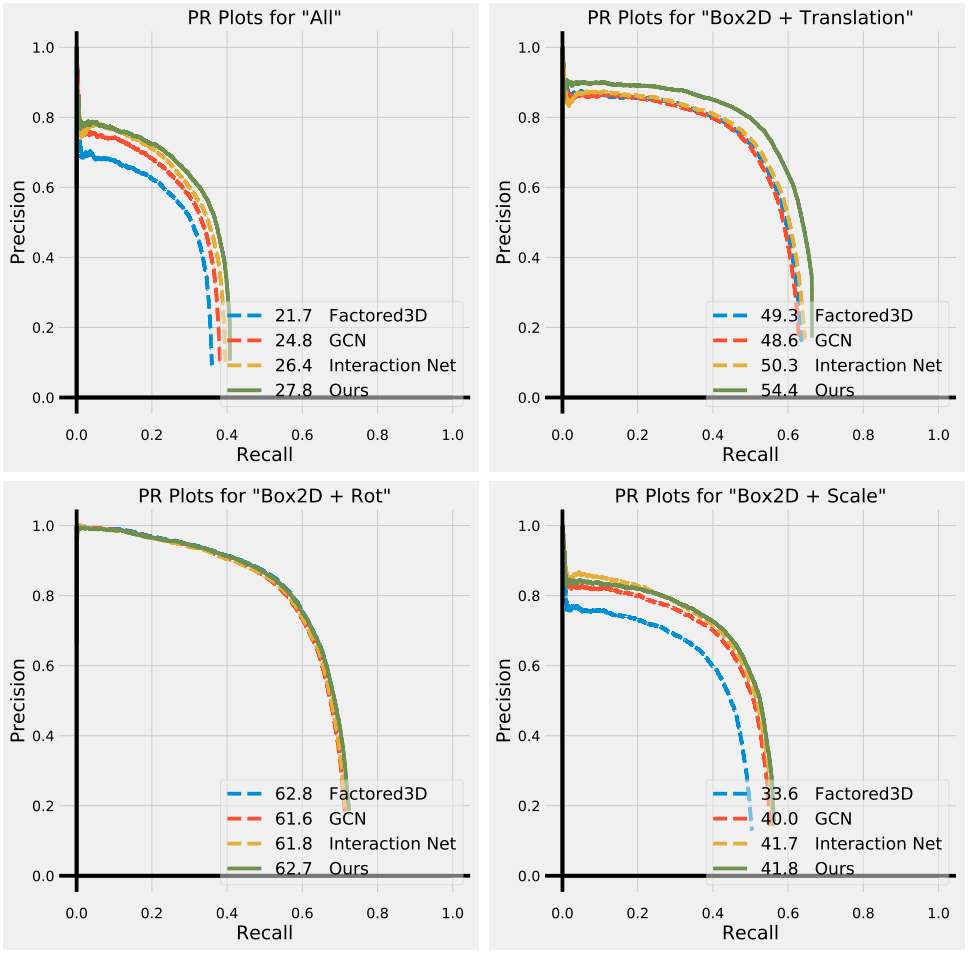

Figure 3.6: We plot the precision-recall (PR) curves for the **detection setting** for SUNCG and also display the mean Average Precision (AP) values in the legend. In each of these curves, we vary the criteria used to determine a true positive. This helps us analyze the relative contribution of each component (translation, rotation, scale) to the final performance.

F.3 Additional Visualizations and Results

We visualize the precision-recall curves in the detection setting using the SUNCG dataset in figure **??**. We also visualize predictions for randomly sampled images in the setting with known bounding boxes in figure **??**.

F.4 Visualization on NYU in Detection Setting

We visualize sample from NYU in the detection setting, and show comparisons with respect to the baseline. Please refer to Figure **??**.

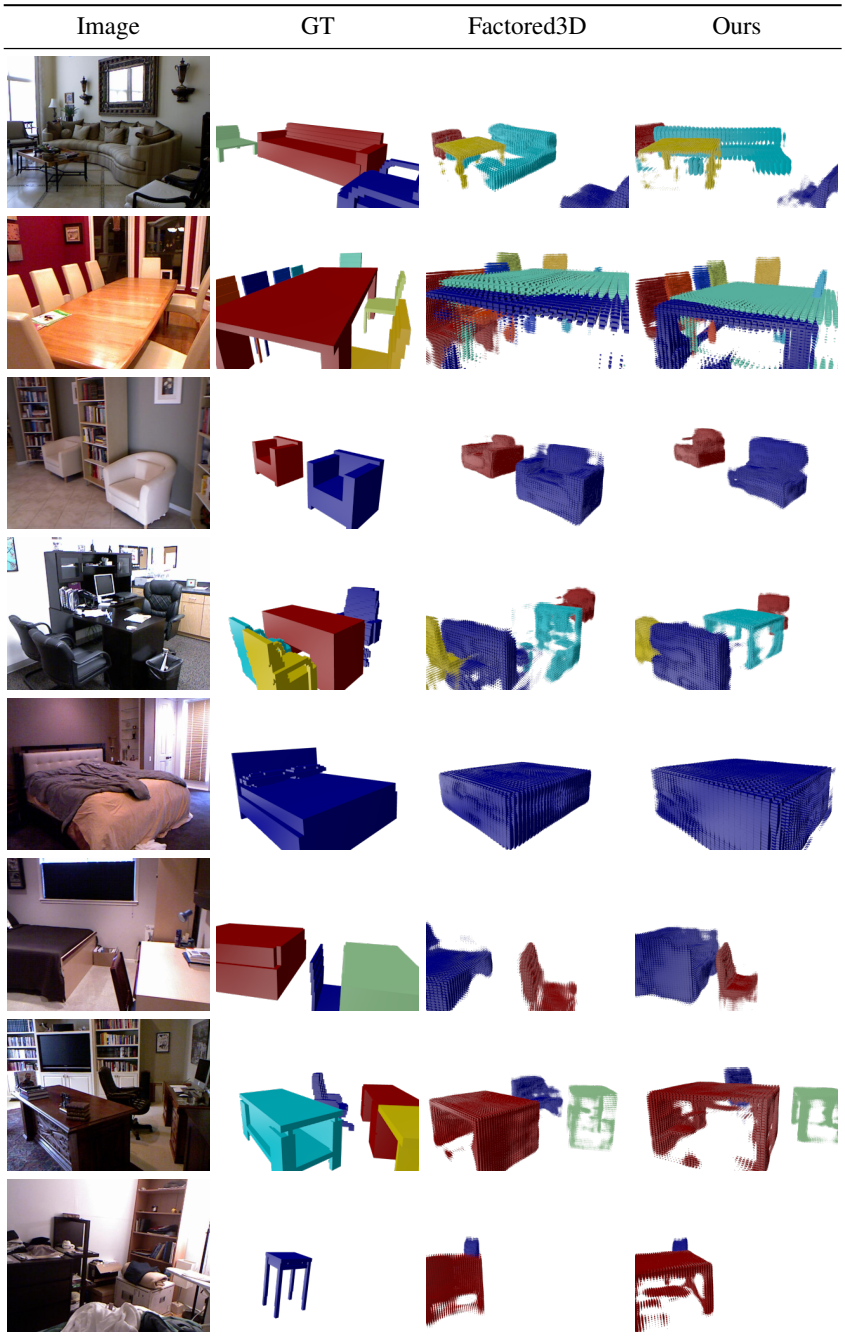

Figure 3.7: Detection Setting for NYU: We visualize sample prediction results in the detection setting Section 4.3 of the main manuscript. We can notice that relative arrangement between objects is better under the Ours column vs Factored3D.

F.5 Factored3D + CRF Details

We implement the CRF model by creating statistical models for relative translation, relative scale, and relative direction for every pair of object categories. We fit a mixture of 10 Gaussian to the data from each pair and modality. At test time we optimize using this prior assuming access to ground truth class labels to choose the appropriate prior. For optimization we use LBFGS [**?**] and we optimize for 1000 iterations for every example. We visualize the outputs for CRF + Factored3D model and compare against the baseline in Figure **??**

Figure 3.8: Factored3D + CRF in GT Box setting: We visualize sample prediction results in the GT Box setting Section 4.2 of the main manuscript for the Factored3D + CRF model. The first two rows show examples where the Factored3D + CRF model does better than Factored3D baseline while the next two rows show the examples where it does worse.

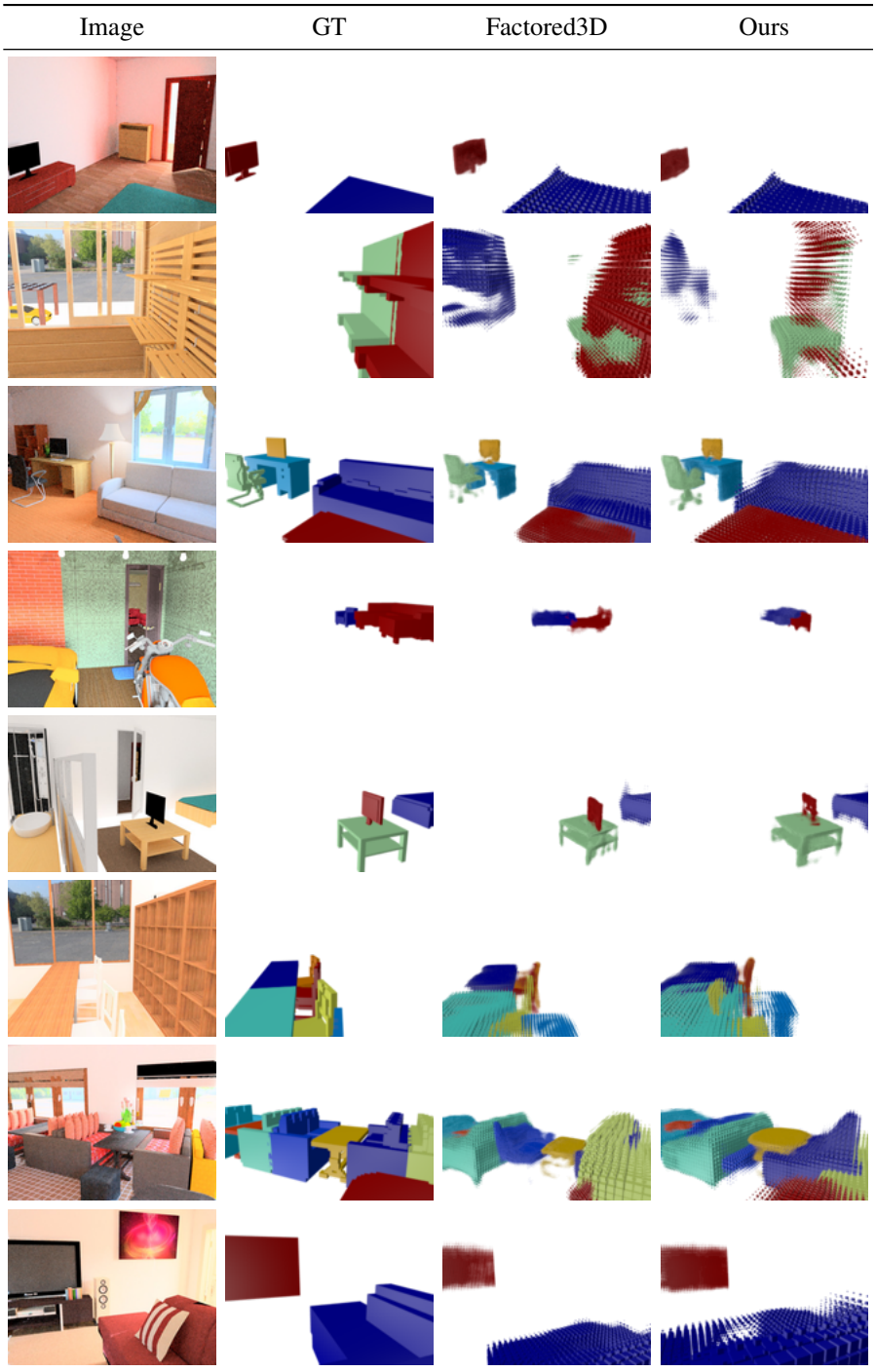

Figure 3.9: We visualize predictions for *randomly sampled images* in the setting with known ground-truth boxes for the SUNCG dataset.

Bibliography

[1] Free3d.com. `http://www.free3d.com`.

[2] R. Alp Güler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[3] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. SCAPE: Shape Completion and Animation of PEople. 2005.

[4] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic. Seeing 3D chairs: exemplar part-based 2D-3D alignment using a large dataset of CAD models. In *CVPR*, 2014.

[5] A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2D-3D alignment via surface normal prediction. In *CVPR*, 2016.

[6] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, et al. Interaction networks for learning about objects, relations and physics. In *NIPS*, 2016.

[7] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *ACM SIGGRAPH*, 1999.

[8] T. J. Cashman and A. W. Fitzgibbon. What shape are dolphins? building 3d morphable models from 2d images. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):232–244, 2013.

[9] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[10] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *CVPR*, 2013.

[11] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker. Universal correspondence network. In *Advances in Neural Information Processing Systems*, pages 2414–2422, 2016.

[12] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016.

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

[14] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[15] S. Fidler, S. Dickinson, and R. Urtasun. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *NIPS*, 2012.

[16] M. Fisher, D. Ritchie, M. Savva, T. Funkhouser, and P. Hanrahan. Example-based synthesis of 3d object arrangements. *ACM Transactions on Graphics (TOG)*, 31(6):135, 2012.

[17] M. Fisher, M. Savva, and P. Hanrahan. Characterizing structural relationships in scenes using graph kernels. *ACM transactions on graphics (TOG)*, 30(4):34, 2011.

[18] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.

[19] R. Girdhar, D. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016.

[20] R. Girshick. Fast r-cnn. In *ICCV*, 2015.

[21] G. Gkioxari, R. Girshick, P. Dollár, and K. He. Detecting and recognizing human-object intaractions. *CVPR*, 2018.

[22] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[23] R. Guo and D. Hoiem. Support surface prediction in indoor scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[24] A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. *ECCV*, 2010.

[25] S. Gupta and J. Malik. Visual semantic role labeling. *arXiv preprint arXiv:1505.04474*, 2015.

[26] A. Guzmán. Decomposition of a visual scene into three-dimensional bodies. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*. ACM, 1968.

[27] J. Gwak, C. B. Choy, A. Garg, M. Chandraker, and S. Savarese. Weakly supervised 3d reconstruction with adversarial constraint. In *3DV*, 2017.

[28] B. Ham, M. Cho, C. Schmid, and J. Ponce. Proposal flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3475–3484, 2016.

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

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

[31] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *CVPR*, 2009.

[32] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[33] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *CVPR*, 2005.

[34] S. Huang, S. Qi, Y. Zhu, Y. Xiao, Y. Xu, and S.-C. Zhu. Holistic 3d scene parsing and reconstruction from a single rgb image. In *European Conference on Computer Vision*, pages 194–211. Springer, 2018.

[35] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *IJCV*, 1990.

[36] E. Insafutdinov and A. Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *Advances in Neural Information Processing Systems*, pages 2807–2817, 2018.

[37] H. Izadinia, Q. Shan, and S. M. Seitz. IM2CAD. In *CVPR*, 2017.

[38] C. Jiang, S. Qi, Y. Zhu, S. Huang, J. Lin, L.-F. Yu, D. Terzopoulos, and S.-C. Zhu. Configurable 3d scene synthesis and 2d image rendering with per-pixel ground truth using stochastic grammars. *International Journal of Computer Vision*, 126(9):920–941, 2018.

[39] Y. Jiang, M. Lim, and A. Saxena. Learning object arrangements in 3d scenes using human context. *arXiv preprint arXiv:1206.6462*, 2012.

[40] A. Kanazawa, D. W. Jacobs, and M. Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261, 2016.

[41] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018.

[42] A. Kar, S. Tulsiani, J. Carreira, and J. Malik. Category-specific object reconstruction from a single image. 2015.

[43] A. Kar, S. Tulsiani, J. Carreira, and J. Malik. Category-specific object reconstruction from a single image. In *CVPR*, 2015.

[44] H. Kato, Y. Ushiku, and T. Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[45] S. Khamis, J. Taylor, J. Shotton, C. Keskin, S. Izadi, and A. Fitzgibbon. Learning an efficient model of hand shape variation from depth images. 2015.

[46] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2307–2314, 2013.

[47] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1857–1865. JMLR. org, 2017.

[48] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[49] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[50] N. Kulkarni, I. Misra, S. Tulsiani, and A. Gupta. 3d-relnet: Joint object and relational network for 3d prediction. 2018.

[51] N. Kulkarni, S. Tulsiani, and A. Gupta. Canonical surface mapping via geometric cycle consistency. 2018.

[52] A. Kundu, Y. Li, and J. M. Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *CVPR*, 2018.

[53] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[54] D. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. *NIPS*, 2010.

[55] Y. Li, H. Su, C. R. Qi, N. Fish, D. Cohen-Or, and L. J. Guibas. Joint embeddings of shapes and images via cnn image purification. *TOG*, 2015.

[56] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing ikea objects: Fine pose estimation. In *ICCV*, 2013.

[57] D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *ICCV*, 2013.

[58] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2011.

[59] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[60] J. L. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2014.

[61] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 2015.

[62] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016.

[63] M. Osadchy, Y. L. Cun, and M. L. Miller. Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research*, 2007.

[64] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis. 6-dof object pose from semantic keypoints. In *ICRA*, 2017.

[65] E. Praun and H. Hoppe. Spherical parametrization and remeshing. In *ACM Transactions on Graphics (TOG)*, 2003.

[66] D. J. Rezende, S. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess. Unsupervised learning of 3d structure from images. 2016.

[67] L. G. Roberts. *Machine Perception of Three-Dimensional Solids*. PhD thesis, MIT, 1963.

[68] I. Rocco, R. Arandjelovic, and J. Sivic. Convolutional neural network architecture for geometric matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[69] I. Rocco, R. Arandjelović, and J. Sivic. End-to-end weakly-supervised semantic alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[70] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[71] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017.

[72] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3D scene structure from a single still image. *TPAMI*, 2009.

[73] T. Schmidt, R. Newcombe, and D. Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2017.

[74] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction for 3d indoor scene understanding. In *CVPR*, 2012.

[75] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[76] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[77] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017.

[78] J. Taylor, R. Stebbing, V. Ramakrishna, C. Keskin, J. Shotton, S. Izadi, A. Hertzmann, and A. Fitzgibbon. User-specific hand modeling from monocular depth sequences. 2014.

[79] J. Thewlis, H. Bilen, and A. Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *Advances in Neural Information Processing Systems*, pages 844–855, 2017.

[80] D. Thompson. *On Growth and Form*. Cambridge Univ. Press, 1917.

[81] S. Tulsiani, A. A. Efros, and J. Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *Computer Vision and Pattern Regognition (CVPR)*, 2018.

[82] S. Tulsiani, S. Gupta, D. Fouhey, A. A. Efros, and J. Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *CVPR*, 2018.

[83] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *CVPR*, 2015.

[84] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. 2017.

[85] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[86] X. Wang, K. He, and A. Gupta. Transitive invariance for self-supervised visual representation learning. In *ICCV*, 2017.

[87] X. Wang, A. Jabri, and A. A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

[88] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman. Single image 3d interpreter network. 2016.

[89] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 75–82. IEEE, 2014.

[90] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. 2016.

[91] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[92] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *CVPR*, 2017.

[93] Y. Zhao and S.-C. Zhu. Image parsing with stochastic scene grammar. In *NIPS*, 2011.

[94] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.

[95] T. Zhou, Y. Jae Lee, S. X. Yu, and A. A. Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1191–1200, 2015.

[96] T. Zhou, P. Krähenbühl, M. Aubry, Q. Huang, and A. A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[97] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

[98] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.