# Learning to Reconstruct 3D Objects

*Abhishek Kar*

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2017-199
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-199.html

December 12, 2017

**Learning to Reconstruct 3D Objects**

by

Abhishek Kar

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Jitendra Malik, Chair
Professor Alexei Efros
Professor Bruno Olshausen

Fall 2017

**Learning to Reconstruct 3D Objects**



Abhishek Kar

## Abstract

Learning to Reconstruct 3D Objects

by

Abhishek Kar
Doctor of Philosophy in Computer Science

University of California, Berkeley
Professor Jitendra Malik, Chair

Ever since the dawn of computer vision, 3D reconstruction has been a core problem, inspiring early seminal works and leading to numerous real world applications. Much recent progress in the field however, has been driven by visual recognition systems powered by statistical learning techniques - more recently with deep convolutional neural networks (CNNs). In this thesis, we attempt to bridge the worlds of geometric 3D reconstruction and learning based recognition by learning to leverage various 3D perception cues from image collections for the task of reconstructing 3D objects.

In Chapter 2, we present a system that is able to learn intra-category regularities in object shapes by building category-specific deformable 3D models from 2D recognition datasets enabling fully automatic single view 3D reconstruction for novel instances. In Chapter 3, we demonstrate how predicting the amodal extent of objects in images and reasoning about their co-occurrences can help us infer their real world heights. Finally, in Chapter 4, we present Learnt Stereo Machines (LSM), an end-to-end learnt framework using convolutional neural networks, which unifies a number of paradigms in 3D object reconstruction- single and multi-view reconstruction, coarse and dense outputs and geometric and semantic reasoning. We will conclude with several promising future directions for learning based 3D reconstruction.

Professor Jitendra Malik
Dissertation Committee Chair

To my grandparents.

# Contents

# Acknowledgments

This dissertation is a product of the affection and guidance of a great many people who contributions to this perhaps outweigh any of mine. First and foremost, I would like to thank my advisor, Jitendra Malik, for teaching me how to pursue impactful research, the numerous history lessons and having my back whenever I faltered. I wouldn't trade advisors for anything in this world. Thank you Alyosha Efros for (re)-introducing me to the treasure that is Berkeley and guiding me through testing times both as a mentor and an academic "brother". Thanks to my quals and thesis committee members Bruno Olshausen and Pieter Abbeel for valuable feedback on my research.

I have had the great fortune to be surrounded by some of the smartest and most empathetic people during my time at Berkeley. Thanks to my co-authors - Shubham Tulsiani who taught me great many things including the art of asking the right questions, João Carreira for his artful writing and Christian Häne for deepening my knowledge in 3D vision. Thanks to everyone in Malik and Efros groups: Saurabh, Bharath, Pablo, Jon Barron, Georgia, Pulkit, Panna, Ross, Deepak, Tinghui, Jun-Yan, Richard, Shiry, David, Evan, Judy and others, for all the discussions in the lab. My ideas and opinions have been greatly shaped by the discussions and I have learned immensely from them. I would also like to thank Angie Abbatecola for never letting the Berkeley bureaucracy bog me down and always being there to help.

My experience during grad school has been made ever so enjoyable by my amazing support group of friends. Thank you Sakshi, Somil, Tejas, Mobin, Smeet and Shiva for being there when the going got tough. Thanks to my group of friends away from home (Berkeley) - Partha, Sanjay and Pramit, for keeping me going. Thanks Varsha for always believing in me and helping me become a better human being over these years. Finally, I would like to thank the three most important people in my life: my father, mother and brother (Amlan). Thank you for giving me the freedom to pursue my dreams, teaching me the immense value of helping others and never letting me feel alone 8000 miles away.

# Chapter 1

# Introduction

Consider looking at a photograph of a chair. We humans have the remarkable capacity of inferring properties about the 3D shape of the chair from this single photograph even if we might not have seen such a chair ever before. A more representative example of our experience though is being in the same physical space as the chair and accumulating information from various viewpoints around it to build up our hypothesis of the chairs 3D shape. How do we solve this complex 2D to 3D inference task? What kind of cues do we use? How do we seamlessly integrate information from just a few views to build up a holistic 3D model of the object?

A vast body of work in computer vision has been devoted to developing algorithms which leverage various cues from images that enable this task of 3D reconstruction. They range from monocular cues such as shading, linear perspective, size constancy *etc.* [1] to binocular [2] and even multi-view stereopsis. The dominant paradigm for integrating multiple views has been to leverage stereopsis, i.e. if a point in the 3D world is viewed from multiple viewpoints, its location in 3D can be determined by triangulating its projections in the respective views. This family of algorithms has led to work on Structure from Motion (SfM) [3, 4, 5] and Multi-view Stereo (MVS) [6, 7] and have been used to produce city-scale 3D models [8] and enable rich visual experiences such as 3D flyover maps [9].

While multi-view 3D reconstruction has focussed on scaling up to larger scenes with smarter optimization schemes and large scale engineering solutions, a complementary line of work in 3D inference has focussed on modelling 3D shapes of objects. A well studied example of such an object category is human faces. A key aspect of this problem is modeling the intraclass variation in object shape as demonstrated in the seminal work of Blanz and Vetter [10]. Recent advances [11, 12] in 3D face modelling have resulted in real-time systems for facial animation [13] and face-based authentication.

Perhaps in contrast to 3D reconstruction, recognition systems in computer vision (object classification, detection, segmentation *etc.*) have improved leaps and bounds in recent times - primarily driven by the success of powerful deep neural networks [14]

in modelling highly complex data distributions. As a result, we now have ready access to cues for 3D perception such as object silhouettes, keypoints, poses as well as the opportunity to borrow ideas from these recognition systems to model rich shape distributions. It is precisely these avenues for 3D reconstruction that this thesis explores - how can recognition help 3D reconstruction, specifically for the task of 3D object reconstruction.

This thesis makes the following contributions in the area of learning-based 3D object reconstruction:

- Building statistical models of 3D shapes for diverse object categories beyond faces and using them in conjunction with recognition techniques for fully automatic single-view 3D reconstruction.
- Enriching the output of object detection systems with real world heights of objects by predicting their amodal extent in scenes.
- Unifying single and multi-view 3D object reconstruction by incorporating geometric constraints in state-of-the-art recognition systems (CNNs)

We begin in Chapter 2 by addressing the problem of fully automatic object localization and reconstruction from a single image. This is both a very challenging and very important problem which has received limited attention due to difficulties in segmenting objects and predicting their poses. We leverage advances in learning convolutional networks for object detection [15], instance segmentation [16] and camera viewpoint prediction [17]. These predictors, while very powerful, are still not perfect given the stringent requirements of 3D shape reconstruction. Thus, we introduce a new class of deformable 3D models that can be robustly fitted to images based on noisy pose and silhouette estimates computed upstream and that can be learned directly from 2D annotations available in existing object detection datasets. These deformable shape models capture top-down information about the major modes of shape variation within a class providing a "low-frequency" estimate of 3D shape. In order to capture fine instance-specific shape details, we fuse it with a high-frequency component recovered from bottom-up shading shading cues.

Armed with shape models for objects, we work towards coherently assembling 3D objects in a scene in Chapter 3. More specifically, we look at the task of enriching current object detection systems with veridical object sizes and relative depth estimates from a single image. There are several technical challenges involved here, such as occlusions, lack of calibration data and the scale ambiguity between object size and distance. Here we propose to tackle these issues by building upon advances in object recognition using large-scale datasets. We first introduce the task of amodal bounding box completion, which aims to infer the the full extent of the objects in an image. We then propose a probabilistic framework for learning category-specific object size distributions from available annotations and leverage these in conjunction

with amodal completions to infer veridical sizes of objects in novel images. Finally, we introduce a focal length prediction approach that exploits scene recognition to overcome inherent scale ambiguities and demonstrate qualitative results on challenging real-world scenes.

In Chapter 4, we move beyond single-view 3D reconstruction and propose a unified framework for single and multi-view object reconstruction with calibrated camera poses within an end-to-end learnt framework. End-to-end learning allows us to implicitly model complex shape distributions using powerful CNNs while conforming to geometric constraints imposed by the camera poses. We show large improvements over learning systems that treat 3D reconstruction as a regression problem and classical multi-view stereo systems on low number of views.

We conclude with a discussion on the limitations of current systems and promising future directions in recognition-based 3D reconstruction.

# Chapter 2

# Category-Specific Deformable 3D Models

Consider the chairs in Figure 2.1. As humans, not only can we infer at a glance that the image contains three chairs, we also construct a rich internal representation of each of them such as their locations and 3D poses. Moreover, we have a guess of their 3D shapes, even though we might never have seen these particular chairs. We can do this because we do not experience this image *tabula rasa*, but in the context of our "remembrance of things past". Previously seen chairs enable us to develop a notion of the 3D shape of chairs, which we can project to the instances in this particular image. We also specialize our representation to these particular instances (e.g. any custom decorations they might have), signalling that both top-down and bottom-up cues influence our percept [19]. In this chapter, we incorporate these principles in a computational framewoek for reconstructing objects given a single image.

The task of reconstructing objects from a single image is a challenging one – a typical image depicts many objects, each possibly belonging to a different object category; an object category, in turn, comprises instances of varying shapes, textures, size *etc.* and any particular instance may be viewed from a different viewpoint. Previous approaches to this problem can be broadly grouped into two paradigms. The paradigm of model-based object reconstruction has reflected varying preferences on model representations. Generalized cylinders [20] resulted in very compact descriptions for certain classes of shapes, and can be used for category level descriptions, but the fitting problem for general shapes is challenging. Polyhedral models [21, 22], which trace back to the early work of Roberts [23], and CAD models [24, 25, 26], cannot perfectly deform into shapes even slightly different from those in training

This chapter is based on joint work with Shubham Tulsiani, João Carreira and Jitendra Malik [18], presented primarily as it appears in the TPAMI 2017 proceedings. Statements throughout the chapter (such as references to "prior work") should be read with this context in mind.

**Figure 2.1:** Example outputs of our system, given a single image of a scene having chairs, a class that the system was exposed to during training. The coloring on the right image signals object-centric depth (we do not aim for globally consistent depths across multiple objects). Blue means close to the camera, red means far from the camera.

data, but given a set of point correspondences can be quite effective for determining approximate instance viewpoints. Some recent methods have proposed using similar instances from a collection of CAD models [27, 28] for non-parametric reconstruction but their applications have been restricted to pre-segmented online product images or recovering 3D from 2.5D object scans [29]. Here we pursue more expressive basis shape models [30, 10, 31] which establish a balance between the two extremes as they can deform but only along class-specific modes of variation.

The alternate paradigm comprises of approaches that target the problem of object reconstruction in a class or object agnostic manner, either implicitly or explicitly using generic learned 3D shape cues [32, 33], or bottom-up cues and the physics of image formation [34, 35] building upon the long tradition of Shape-from-X, which traces back to seminal work by Horn [36]. These methods, while quite general, have not yet been demonstrated for 3D reconstruction – as opposed to 2.5D – and typically assume known object segmentation [35]. Some recent approaches have demonstrated the use of supervised learning techniques to implcitly learn generic cues to predict depth maps [37] and surface normals [38, 39] but these have primarily focused on inferring scene-level information which differs from our goal of perceiving the shape of objects.

In this chapter, we combine both these reconstruction paradigms - we obtain top-down shape information from our model-based reconstruction approach and complement it with bottom-up shape information obtained via an intrinsic image decomposition method. Crucially, in contrast to previous work (e.g. [35, 40, 41]), we do not require perfect knowledge of object localization and pose as our reconstruction is driven by automatic figure-ground object segmentations and viewpoint estimations.

The framework we propose to reconstruct the objects present in an image is out-

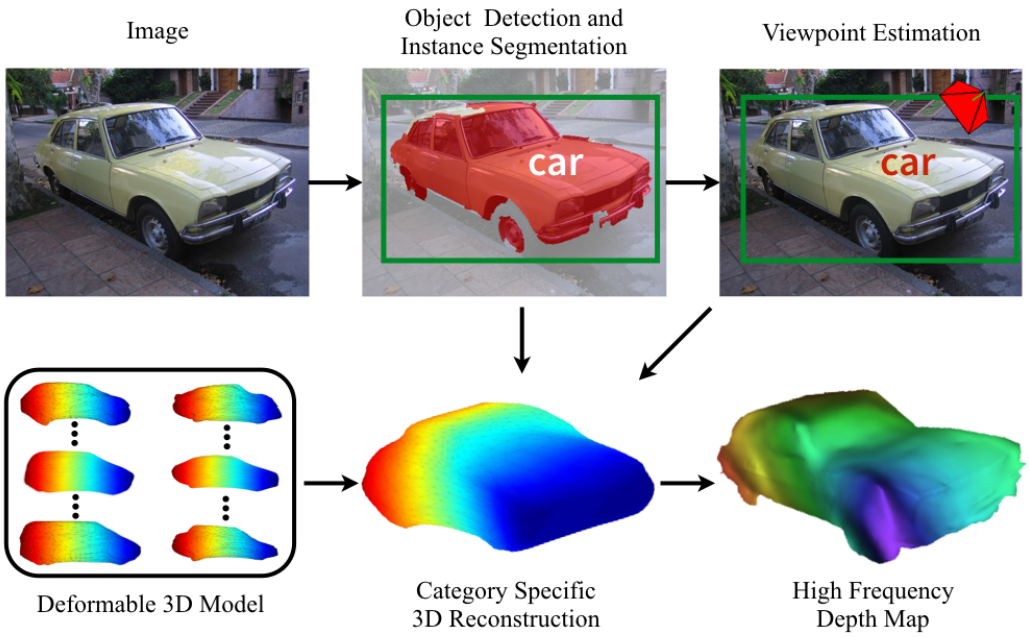


**Figure 2.2:** Overview of our full reconstruction method. We leverage estimated instance segmentations and predicted viewpoints to generate a full 3D mesh and a high frequency 2.5D depth map for each object in the image.

lined in Figure 2.2. As a first step, we leverage the recent progress made by the computer vision community in object detection [15] and instance segmentation [42, 16] to identify and localize objects in the image. For each object, we also predict a viewpoint in the form of three euler angles. We then use our learned deformable 3D shape models in conjunction with the viewpoint and localization information to produce a "top-down" 3D reconstruction for the object guided primarily by category level cues. Finally, we infuse our 3D shape with high frequency local shape cues to obtain our end result - a rich 3D reconstruction of the object. We briefly outline each of the components required for the above proposed framework.

**Learning Deformable 3D Models.** As noted earlier, previously seen objects allow us to develop a notion of 3D shape which informs inference for new instances. We present an algorithm that can build category-specific deformable shape models from just images with 2D annotations (segmentation masks and a small set of keypoints) present in modern computer vision datasets (e.g. PASCAL VOC [43]). These learnt shape models and deformations allow us to robustly infer shape while capturing intra-class shape variation.

**Object Shape Recovery.** Given an object's category, approximate localization and viewpoint, we obtain a 3D reconstruction for the corresponding object using the learned category-specific deformable shape model. We complement the top-down

shape inferred via this inference with a bottom-up module that further refines our shape estimate for a particular instance. This framework allows us to capture the coarse as well as fine level shape details for objects from a single image.

This chapter is organized as follows: in Section 2.1 we describe our model learning pipeline where we estimate camera parameters for all training objects (Section 2.1.1) followed by our shape model formulation (Section 2.1.2) to learn 3D models. Section 2.2 describes our testing pipeline where we leverage our learnt models alongwith object recognition systems (detection [15], segmentation [16], pose estimation [17]) to reconstruct novel instances without assuming any annotations. We quantitatively evaluate the various components of our approach in Section 2.3 and provide sample reconstructions in the wild.

## 2.1 Learning Deformable 3D Models

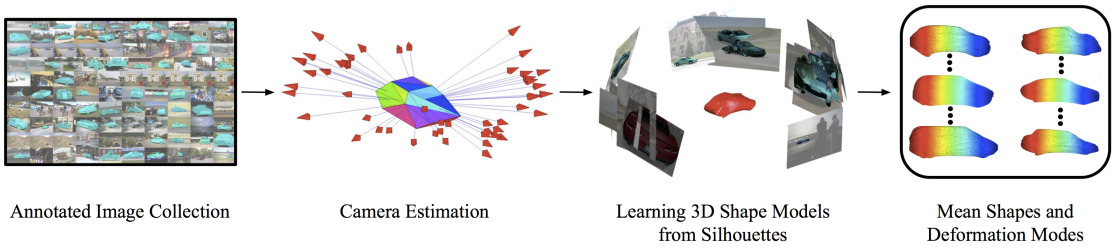


**Figure 2.3:** Overview of our training pipeline. We use an annotated image collection to estimate camera projection parameters which we then use along with object silhouettes to learn 3D shape models. Our learnt shape models, as illustrated in the rightmost figure are capable of deforming to capture intra-class shape variation.

We are interested in learning 3D shape models that can be robustly aligned to noisy object segmentations by incorporating top-down class-specific knowledge of how shapes from the class typically project onto the image. We want to learn such models from just 2D training images, aided by ground truth segmentations and a few keypoints, similar to [40]. Our approach operates by first estimating the projection parameters (camera) for all objects in a class using a structure-from-motion approach, followed by optimizing over a deformation basis of representative 3D shapes that best explain all silhouettes, conditioned on the estimated cameras. We describe these two stages of model learning in the following subsections. Figure 2.3 illustrates this training pipeline of ours.

### 2.1.1 Camera Estimation

We use the framework of NRSfM [44] to jointly estimate the projection parameters (rotation, translation and scale) for all training instances in each class. Originally

proposed for recovering shape and deformations from video [45, 46, 47, 44], NRSfM is a natural choice for camera estimation from sparse correspondences as intra-class variation may become a confounding factor if not modeled explicitly. However, the performance of such algorithms has only been explored on simple categories, such as SUV's [48] or flower petal and clown fish [49]. Closer to our work, Hejrati and Ramanan [50] used NRSfM on a larger class (cars) but need a predictive detector to fill-in missing data (occluded keypoints) which we do not assume to have here.

We closely follow the EM-PPCA formulation of Torresani *et al.* [46] and propose a simple extension to the algorithm that incorporates silhouette information in addition to keypoint correspondences to robustly recover cameras and shape bases. Energies similar to ours have been proposed in the shape-from-silhouette [51] and rigid structure-from-motion [40] literature but, to the best of our knowledge, not in conjunction with NRSfM.

**NRSfM Model Formulation.** We are provided with an annotated training set $T : \{(O_n, P_n)\}_{n=1}^{N}$, where $O_n$ is the instance silhouette and $P_n \in \mathbb{R}^{2\times K}$ denotes the annotated keypoint coordinates, possibly with missing entries (occluded/truncated keypoints). The annotated keypoints $P_n$ are projections of the underlying 3D points $W_n \in \mathbb{R}^{3\times K}$ via the projection function $\pi_n$. In the NRSfM model, the space of 3D keypoint locations $W_n$ is parametrized linearly and the projection function is assumed to be weakly orthographic *i.e.* $\pi_n \equiv (c_n, R_n, T_n)$, where $c_n$ represents scale, $R_n \in \mathbb{R}^{2\times 3}$ denotes rotation and $T_n \in \mathbb{R}^{1\times 2}$ corresponds to 2D translation. Our goal is to infer the camera parameters $(c_n, R_n, T_n)$ as well as 3D keypoint locations $W_n$ for all instances in the annotated training set.

Formally, our adaptation of the NRSfM algorithm in [46] corresponds to maximizing the likelihood of the following model:

$$\begin{aligned} P_n &= c_n R_n W_n + 1^T T_n + N_n \\ W_n &= \bar{W} + \sum_{k=1}^{B} U_b z_{nb} \\ z_n &\sim \mathcal{N}(0, I), \quad N_n^k \sim \mathcal{N}(0, \sigma^2 I) \end{aligned} \tag{2.1}$$

$$\begin{aligned} &\text{subject to:} \quad R_n R_n^T = I_2 \\ &\sum_{k=1}^{K} C_n^{mask}(p_{k,n}) = 0, \quad \forall n \in \{1, \cdots, N\} \end{aligned} \tag{2.2}$$

Here, the (partially) observed keypoint locations $P_n$ are assumed to be the projection under $\pi_n \equiv (c_n, R_n, T_n)$ of the 3D shape $W_n$ with white noise $N_n$. The shape is parameterized as a factored Gaussian with a mean shape $\bar{W}$, $B$ basis vectors $[U_1, U_2, \cdots, U_B] = U$ and latent deformation parameters $z_n$. Our key modification

is constraint in Eq. 2.2 where $C_n^{mask}$ denotes the Chamfer distance field of the $n^{th}$ instance's binary mask and says that all keypoints $p_{k,n}$ of instance $n$ should lie inside its binary mask. We observed that this results in more accurate cameras as well as more meaningful shape bases learnt from the data.

**Learning.** The likelihood of the above model is maximized using the EM algorithm. Missing data (occluded keypoints) is dealt with by "filling-in" the values using the forward equations after the E-step. The algorithm computes shape parameters $\{\bar{W}, U\}$, rigid body transformations $\{c_n, R_n, T_n\}$ as well as the deformation parameters $\{z_n\}$ for each training instance $n$. In practice, we augment the data using horizontally mirrored images to exploit bilateral symmetry in the object classes considered. We also precompute the Chamfer distance fields for the whole set to speed up computation. As shown in Figure 2.4, NRSfM allows us to reliably predict cameras while being robust to intraclass variations.

**Figure 2.4:** NRSfM camera estimation: Estimated cameras visualized using a 3D car wireframe.

### 2.1.2 3D Basis Shape Model Learning

Equipped with camera projection parameters and keypoint correspondences (lifted to 3D by NRSfM) on the whole training set, we proceed to build deformable 3D shape models from object silhouettes within the same class. 3D shape reconstruction from multiple silhouettes projected from a single object in calibrated settings has

been widely studied. Two prominent approaches are visual hulls [52] and variational methods derived from snakes [53, 54] which deform a surface mesh iteratively until convergence. Some works have extended variational approaches to handle categories [41, 55] but typically require some form of 3D annotations to bootstrap models. A contemporary visual-hull based approach [40] requires only 2D annotations as we do for class-based reconstruction and it was successfully demonstrated on PASCAL VOC but does not serve our purpose as it makes strong assumptions about the accuracy of the segmentation and will fill entirely any segmentation with a voxel layer. In contrast, we build parametric shape models for categories that compactly capture intra class shape variations. The benefits of having a model of 3D shape are manifold: 1) we are more robust to noisy inputs (silhouettes and pose) allowing us to pursue reconstruction in a fully automatic setting and 2) we can potentially sample novel shapes from an object category.

**Shape Model Formulation.** We model our category shapes as a deformable point cloud - one per object class. As in the NRSfM model, we use a linear combination of basis vectors to model these deformations. Note that we learn such models from silhouettes and this is what enables us to learn deformable models without relying on point correspondences between scanned 3D exemplars [56].

The annotated training set $T : \{(O_n, P_n)\}_{n=1}^{N}$, where $O_n$ is the instance silhouette and $P_n \in \mathbb{R}^{2\times K}$ denotes the annotated keypoint coordinates, is augmented after NRSfM to contain $\pi_n$ (the projection function from world to image coordinates) and $W_n$ (3D coordinates for a small set of keypoints). Our shape model $M = (\overline{S}, V)$ comprises of a mean shape $\overline{S}$ and deformation bases $V = \{V_1, ., V_K\}$ learnt from the augmented training set $T : \{(O_n, \pi_n, W_n)\}_{n=1}^{N}$. Note that the $\pi_i$ we obtain using NRSfM corresponds to orthographic projection but our algorithm could handle perspective projection as well.

In addition to the above, we use the following notations – $\pi(S)$ corresponds to the 2D projection of shape $S$, $C^{mask}$ refers to the Chamfer distance field of the binary mask of silhouette $O$ and $\Delta^k(p; Q)$ is defined as the squared average distance of point $p$ to its $k$ nearest neighbors in set $Q$.

**Energy Formulation.** We formulate our objective function primarily based on image silhouettes. For example, the shape for an instance should always project within its silhouette and should agree with the keypoints (lifted to 3D by NRSfM ). We capture these by defining corresponding energy terms as follows:

**Silhouette Consistency.** Silhouette consistency simply enforces the predicted shape for an instance to project inside its silhouette. This can be achieved by penalizing the points projected outside the instance mask by their distance from the silhouette (*i.e.* squared distance to the closest silhouette point). In our $\Delta$ notation it can be

written as follows:

$$E_s(S, O, \pi) = \sum_{C^{mask}(p)>0} \Delta^1(p; O) \tag{2.3}$$

**Silhouette Coverage.** Using silhouette consistency alone would just drive points projected outside in towards the silhouette. This wouldn't ensure though that the object silhouette is "filled" - i.e. there might be overcarving. We deal with it by having an energy term that encourages points on the silhouette to pull nearby projected points towards them. Formally, this can be expressed as:

$$E_c(S, O, \pi) = \sum_{p \in O} \Delta^m(p; \pi(S)) \tag{2.4}$$

**Keypoint Consistency.** Our NRSfM algorithm provides us with sparse 3D keypoints along with camera projection parameters. We use these sparse correspondences on the training set to deform the shape to explain these 3D points. The corresponding energy term penalizes deviation of the shape from the 3D keypoints $W$ for each instance. Specifically, this can be written as:

$$E_{kp}(S, W) = \sum_{\kappa \in W} \Delta^m(\kappa; S) \tag{2.5}$$

**Local Consistency.** In addition to the above data terms, we use a simple shape regularizer to restrict arbitrary deformations by imposing a quadratic deformation penalty between every point and its neighbors. We also impose a similar penalty on deformations to ensure local smoothness. The $\delta$ parameter represents the mean squared displacement between neighboring points and it encourages all faces to have similar size. Here $V_{ki}$ is the $i^{th}$ point in the $k^{th}$ basis.

$$E_l(\bar{S}, V) = \sum_i \sum_{j \in N(i)} ((\|\bar{S}_i - \bar{S}_j\| - \delta)^2 + \sum_k \|V_{ki} - V_{kj}\|^2) \tag{2.6}$$

**Normal Smoothness.** Shapes occurring in the natural world tend to be locally smooth. We capture this prior on shapes by placing a cost on the variation of normal directions in a local neighborhood in the shape. Our normal smoothness energy is formulated as

$$E_n(S) = \sum_i \sum_{j \in N(i)} (1 - \vec{\mathcal{N}}_i \cdot \vec{\mathcal{N}}_j) \tag{2.7}$$

Here, $\vec{\mathcal{N}}_i$ represents the normal for the $i^{th}$ point in shape $S$ which is computed by fitting planes to local point neighborhoods. Our prior essentially states that local

point neighborhoods should be flat. Note that this, in conjunction with our previous energies automatically enforces the commonly used prior that normals should be perpendicular to the viewing direction at the occluding contour [57].

Our total energy is given in equation Eq. 2.8. In addition to the above smoothness priors we also penalize the $L_2$ norm of the deformation parameters $\alpha_i$ to prevent unnaturally large deformations.

$$E_{tot}(\bar{S}, V, \alpha) = E_l(\bar{S}, V) + \sum_i (E_s^i + E_{kp}^i + E_c^i + E_n^i + \sum_k (\|\alpha_{ik} V_k\|_F^2)) \tag{2.8}$$

**Learning.** We solve the optimization problem in equation Eq. 2.9 to obtain our shape model $M = (\bar{S}, V)$. The mean shape and deformation basis are inferred via block-coordinate descent on $(\bar{S}, V)$ and $\alpha$ using sub-gradient computations over the training set. We restrict $\|V_k\|_F$ to be a constant to address the scale ambiguity between $V$ and $\alpha$ in our formulation. In order to deal with imperfect segmentations and wrongly estimated keypoints, we use truncated versions of the above energies that reduce the impact of outliers. The mean shapes learnt using our algorithm for 9 rigid categories in PASCAL VOC are shown in Figure 2.5. Note that in addition to representing the coarse shape details of a category, the model also learns finer structures like chair legs and bicycle handles, which become more prominent with deformations.

$$\begin{aligned} \min_{\bar{S}, V, \alpha} \quad & E_{tot}(\bar{S}, V, \alpha) \\ \text{subject to:} \quad & S^i = \bar{S} + \sum_k \alpha_{ik} V_k \end{aligned} \tag{2.9}$$

Our training objective is highly non-convex and non-smooth and is susceptible to initialization. We follow the suggestion of [53] and initialize our mean shape with a soft visual hull computed using all training instances. The deformation bases and deformation weights are initialized randomly.

**Implementation Details.** The gradients involved in our optimization for shape and projection parameters are extremely efficient to compute. We use approximate nearest neighbors computed using k-d tree to implement silhouette coverage, keypoint consistency gradients and leverage Chamfer distance fields for obtaining silhouette consistency gradients. Our overall computation takes only about 15 min to learn a deformable shape model for an object category with about 500 annotated examples.

## 2.2 Reconstruction in the Wild

Given an image, our goal is to reconstruct the depicted objects. As the initial step, we use existing state-of-the-art systems [42] to detect and segment the objects

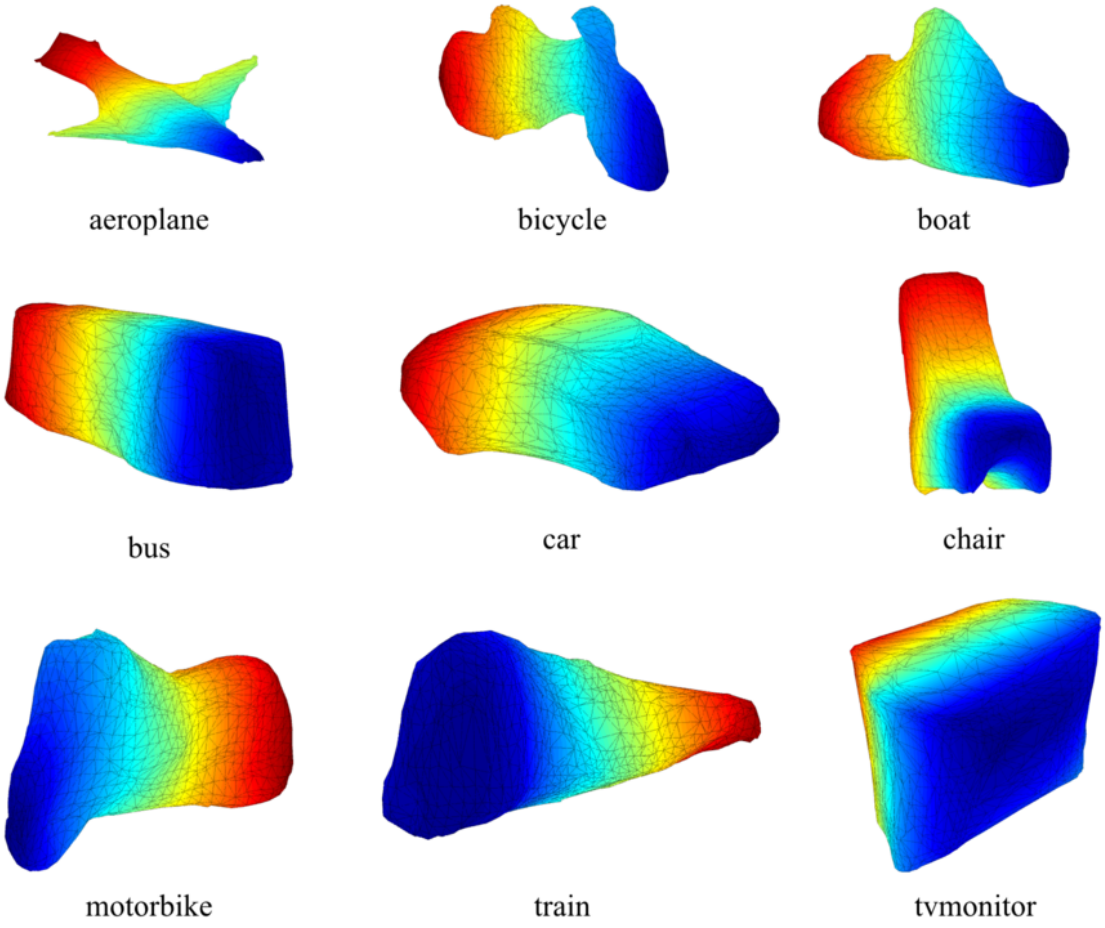


**Figure 2.5:** Mean shapes learnt for rigid classes in PASCAL VOC obtained using our basis shape formulation. Color encodes depth when viewed frontally.

present in the image. We then proceed to individually reconstruct each of the detected objects. We approach the problem of reconstructing these objects from the big picture downward - like a sculptor first hammering out the big chunks and then chiseling out the details. We infer their coarse 3D poses and use these along with the predicted instance segmentations to fit our top-down shape models to obtain a coarse top-down shape (Section 2.2.1). Finally, we recover high frequency shape details from shading cues present in the image (Section 2.2.2).

### 2.2.1 Category Specific Shape Inference

We have at our disposal category-level deformable shape models which can be driven by data-specific and shape-prior based energy terms to infer an object's shape. Recall that the proposed energy terms (Section 2.1.2), in particular silhouette consis-

tency ($E_s(S, O, \pi)$) and silhouette coverage ($E_c(S, O, \pi)$) depend on a known object silhouette $O$ and camera projection $\pi$. We first describe how we estimate $O, \pi$ and then formulate an optimization problem to infer object shape $S$.

**Initialization.** Given an object detection along with its predicted instance segmentation, we use the largest connected component in the predicted segmentation to obtain the object silhouette $O$. We use the viewpoint prediction system described from [17] to predict the viewpoint for the detected object, thereby obtaining the camera rotation $R$. Our learnt models are at a canonical bounding box scale - all objects are first resized to a particular width during training. Given the predicted bounding box, we scale the learnt mean shape accordingly and obtain camera scale $c$. The translation $T$ is initialized to be the center of the predicted bounding box. These provide us an initial estimate of the camera parameters $\pi_0 \equiv (c, R, T)$.

**Formulation.** We want to infer a shape that best explains the observed object silhouette, respects generic shape priors (smoothness, continuity) and lies on the linear manifold of category-level shapes. Note that, unlike model learning phase, we do not have access to annotated keypoint locations and thus do not enforce the reconstruction to explain any keypoint locations. These observations are incorporated by the reconstruction energy defined in (using $E_s, E_c, E_n$ defined in Section 2.1.2).

$$E_r = E_s + E_c + E_n \tag{2.10}$$

In addition to inferring the instance shape, we also observe that the initial camera estimate $\pi_0$ is only approximate as the $R$ is predicted upto a dicretization and $c, T$ are initialized coarsely. To alleviate this, we treat the camera parameters $\pi$ as optimization variables. We further add regularizers to enforce the prior that shape deformation should be small and the the estimated camera should not deviate significantly from the initial camera estimate $\pi_0$ . Our final optimization for inferring the object reconstruction is given in Eq. 2.11.

$$\begin{aligned} \min_{\alpha,\pi} \quad & E_r(S, \pi) + \delta(\pi, \pi_0) + \sum_k (\|\alpha_k V_k\|_F^2)) \\ \text{subject to:} \quad & S = \bar{S} + \sum_k \alpha_k V_k \end{aligned} \tag{2.11}$$

**Inference.** In the above optimization, we first set the optimization variables $\alpha$ and $\pi$ to 0 and $\pi_0$ respectively. We then solve the above minimization for the deformation weights $\alpha$ as well as all the camera projection parameters $\pi$ (scale, translation and rotation) by optimizing Eq. 2.9 using block-coordinate descent (alternately optimizing $\pi$ and $\alpha$). The resulting output from the minimization provides us the projection

| | | aero | bike | boat | bus | car | chair | mbike | sofa | train | tv | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Mesh** | Ours | **1.72** | **1.78** | 3.01 | **1.90** | 1.77 | **2.18** | 1.88 | **2.13** | **2.39** | **3.28** | **2.20** |
| | Carvi [40] | 1.87 | 1.87 | **2.51** | 2.36 | **1.41** | 2.42 | **1.82** | 2.31 | 3.10 | 3.39 | 2.31 |
| | Puff [58] | 3.30 | 2.52 | 2.90 | 3.32 | 2.82 | 3.09 | 2.58 | 2.53 | 3.92 | 3.31 | 3.03 |
| **Depth** | Ours | **9.51** | **9.27** | 17.20 | **12.71** | 9.94 | **7.78** | 9.61 | **13.70** | 31.58 | 8.78 | **13.01** |
| | Carvi [40] | 10.05 | 9.28 | **15.06** | 18.51 | **8.14** | 7.98 | **9.38** | 13.71 | **31.25** | **8.33** | 13.17 |
| | SIRFS [35] | 13.52 | 13.79 | 20.78 | 29.93 | 22.48 | 18.59 | 16.80 | 18.28 | 40.56 | 20.18 | 21.49 |

**Table 2.1:** Studying the quality of our learnt 3D models: comparison between our method and [40, 58] using ground truth keypoints and masks on PASCAL VOC.

parameters $\pi$ as well as the inferred 3D shape $S = \bar{S} + \sum_k \alpha_k V_k$. We use the efficient implementations of energy gradients described earlier and consequently, our overall computation takes only about 2 sec to reconstruct a novel instance using a single CPU core.

### 2.2.2 Bottom-up Shape Refinement

The above optimization results in a top-down 3D reconstruction based on the category-level models, inferred object silhouette, viewpoint and our shape priors. We propose an additional processing step to recover high frequency shape information by adapting the intrinsic images algorithm of Barron and Malik [35, 57], SIRFS, which exploits statistical regularities between shapes, reflectance and illumination Formally, SIRFS is formulated as the following optimization problem:

$$\underset{Z,L}{\text{minimize}} \quad g(I - S(Z,L)) + f(Z) + h(L) \tag{2.12}$$

where $R = I - S(Z,L)$ is a log-reflectance image, $Z$ is a depth map and $L$ is a spherical-harmonic model of illumination. $S(Z,L)$ is a rendering engine which produces a log shading image with the illumination $L$. $g$, $f$ and $h$ are the loss functions corresponding to reflectance, shape and illumination respectively.

We incorporate our current coarse estimate of shape into SIRFS through an additional loss term:

$$f_o(Z,Z') = \sum_i ((Z_i - Z'_i)^2 + \epsilon^2)^{\gamma_o} \tag{2.13}$$

where $Z'$ is the initial coarse shape and $\epsilon$ a parameter added to make the loss differentiable everywhere. We obtain $Z'$ for an object by rendering a depth map of our fitted 3D shape model which guides the optimization of this highly non-convex cost function. The outputs from this bottom-up refinement are reflectance, shape and illumination maps of which we retain the shape.

## 2.3 Experiments

We have presented several contributions towards the goal of object reconstruction from a single image – Section 2.1 proposed a method to learn deformable 3D models from an annotated image set and Section 2.2 put forward a framework for reconstructing objects from a single image. Our goal in the experiments was to empirically evaluate and qualitatively demonstrate the efficacy of each of these contributions.

We first examine the quality and expressiveness of our learned 3D models by evaluating how well they matched the underlying 3D shapes of the training data (Section 2.3.1). We then study their sensitivity of obtained reconstructions when fit to images using noisy automatic segmentations and pose predictions (Section 2.3.2) and finally present qualitative results for reconstructions from a single image (Section 2.3.3).

### 2.3.1 Quality of Learned 3D Models

The first question we address is whether the category-specific shape models we learn for each object class (Section 2.1) using an annotated image collection correctly explain the underlying 3D object shape for these annotated instances. Note that while it is not our final goal, this is itself a very challenging task - we have to obtain a dense 3D reconstruction for annotated images using just silhouettes and sparse keypoint correspondences. Contemporary work by Vicente *et al.* [40] addressed this task of 'lifting' an annotated image collection to 3D and we compare the performance of our model learning stage against their approach. We also incorporate category-agnostic shape inflation [58] and intrinsic image [57] methods as baselines. The evaluation metrics, dataset and results are described below.

**Dataset.** We consider images from the challenging PASCAL VOC 2012 dataset [43] which contain objects from the 10 rigid object categories (as listed in Table 2.1). We use the publicly available ground truth class-specific keypoints [59] and object segmentations [60] to learn category-specific shape models for each class. We learn and fit our 3D models on the whole dataset (no train/test split), following the setup of Vicente *et al.* [40].

Since ground truth 3D shapes are unavailable for PASCAL VOC and most other detection datasets, we evaluated the quality of our learned 3D models on the next best thing we managed to obtain: the PASCAL3D+ dataset [61] which has up to 10 3D CAD models for the rigid categories in PASCAL VOC. PASCAL3D+ provides between 4 different models for "tvmonitor" and "train" and 10 for "car" and "chair". The subset of PASCAL we considered after filtering occluded instances had between 70 images for "sofa" and 500 images for classes "aeroplanes" and "cars".

**Metrics.** We quantify the quality of our 3D models by comparing against the PASCAL 3D+ models using two metrics - 1) a mesh error metric computed as the Hausdorff distance between the ground truth and predicted mesh after translating both to the origin and normalizing by the diagonal of the tightest 3D bounding box of the ground truth mesh [62] and 2) a depth map error to evaluate the quality of the reconstructed visible object surface, measured as the mean absolute distance between reconstructed and ground truth depth:

$$Z\text{-MAE}(\hat{Z}, Z^*) = \frac{1}{n \cdot \gamma} \min_{\beta} \sum_{x,y} |\hat{Z}_{x,y} - Z^*_{x,y} - \beta| \tag{2.14}$$

where $\hat{Z}$ and $Z^*$ represent predicted and ground truth depth maps respectively. Analytically, $\beta$ can be computed as the median of $\hat{Z} - Z^*$ and $\gamma$ is a normalization factor to account for absolute object size for which we use the bounding box diagonal. Note that our depth map error is translation and scale invariant.

**Results.** We report the performance of our model learning approach in Table 2.1. Here, SIRFS denotes a state-of-the art intrinsic image decomposition method and Puffball [58] denotes a shape-inflation method for reconstruction. Carvi denotes the method by Vicente *et al.* [40] which is specifically designed for the task of reconstructing an annotated image collection as their visual hull based reconstruction technique makes strong assumptions regarding the accuracy of the object mask and predicted viewpoint.

We observe that category-agnostic methods – Puffball[58] and SIRFS[35, 57] – consistently perform worse on the benchmark by themselves as they use generic priors to reconstruct each image individually and cannot reason over the image collection jointly. Our model learning performs comparably to the specialized approach of Vicente *et al.*– we demonstrate competitive, if not better, performance on both benchmarks with our models showing greater robustnes to perspective foreshortening effects on "trains" and "buses". Certain classes like "boat" and "sofa" are especially hard because of large intra-class variance and data sparsity respectively.

### 2.3.2 Sensitivity Analysis for Recognition based Reconstruction

Our primary goal is to reconstruct objects in an image automatically. Towards this goal, we study the performance of our system when relaxing the availability of various expensive annotations of the form of keypoint correspondences or instance segmentations.

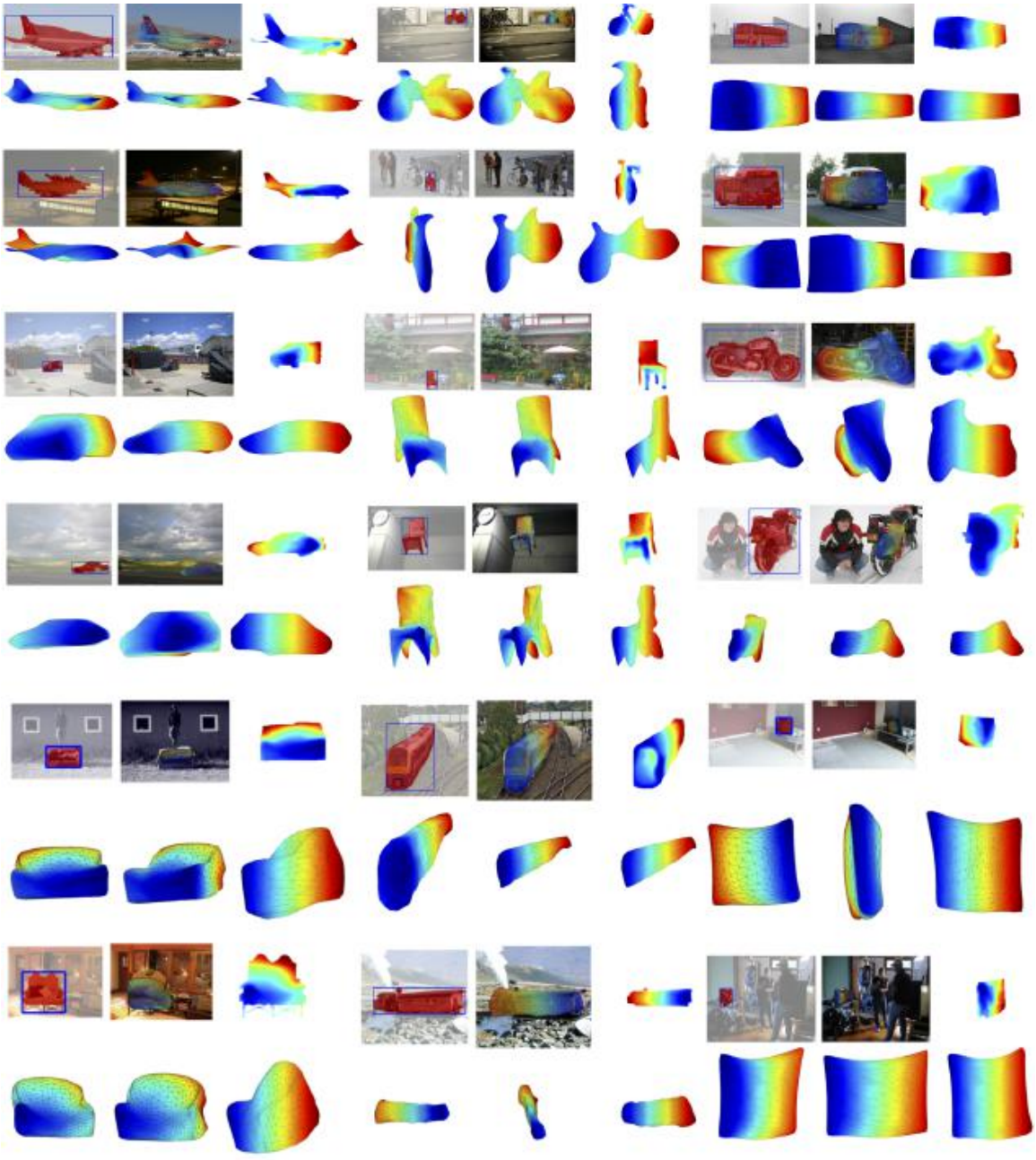

**Figure 2.6:** Fully automatic reconstructions on detected instances (0.5 IoU with ground truth) using our models on rigid categories in PASCAL VOC. We show our instance segmentation input, the inferred shape overlaid on the image, a 2.5D depth map (after the bottom-up refinement stage), the mesh in the image viewpoint and two other views. It can be seen that our method produces plausible reconstructions which is a remarkable achievement given just a single image and noisy instance segmentations. Color encodes depth in the image coordinate frame (blue is closer). More results can be found at `https://goo.gl/MgVQzZ`.

| | | aero | bike | boat | bus | car | chair | mbike | sofa | train | tv | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Mesh** | KP+Mask | 1.77 | 1.85 | 3.68 | 1.90 | 1.80 | 2.26 | 1.83 | 6.86 | 2.69 | 3.40 | 2.80 |
| | KP+SDS | 1.75 | 1.89 | 3.71 | 1.87 | 1.75 | 2.27 | 1.84 | 6.56 | 2.76 | 3.39 | 2.78 |
| | PP+SDS | 1.84 | 2.02 | 4.59 | 1.86 | 1.88 | 2.41 | 2.01 | 7.30 | 2.74 | 3.27 | 2.99 |
| | Puff [58] | 3.31 | 2.49 | 2.95 | 3.40 | 2.87 | 3.09 | 2.65 | 2.73 | 3.91 | 3.33 | 3.07 |
| **Depth** | KP+Mask | 9.83 | 9.95 | 21.07 | 12.80 | 10.07 | 9.10 | 9.98 | 29.39 | 25.70 | 9.85 | 14.77 |
| | KP+SDS | 9.95 | 10.35 | 20.11 | 13.06 | 10.49 | 9.24 | 10.61 | 27.94 | 26.13 | 10.10 | 14.80 |
| | PP+SDS | 11.42 | 11.25 | 21.93 | 22.04 | 13.69 | 10.27 | 11.71 | 26.76 | 34.92 | 9.88 | 17.39 |
| | SIRFS [35] | 13.58 | 14.48 | 19.64 | 30.14 | 22.60 | 20.12 | 16.81 | 21.54 | 41.40 | 23.67 | 22.40 |

**Table 2.2:** Ablation study for our method assuming/relaxing various annotations at test time on objects in PASCAL VOC. As can be seen, our method degrades gracefully with relaxed annotations. Note that these experiments are in a train/test setting and numbers will differ from Table 2.1. Please see text for more details.

**Dataset and Metrics.** The reconstruction error metrics for measuring mesh and depth error are the same as described previously (Section 2.3.1). The segmentation, keypoint annotations for learning and the mesh annotations for evaluation are also similarly obtained. However, for the sensitivity analysis, we introduce a train/test split since the recognition components used for instance segmentation and viewpoint estimation are trained on the PASCAL VOC train set. We therefore train our category-shape models on only the subset of the data corresponding to PASCAL VOC train set. We then reconstruct the held out objects in the PASCAL validation set and report performance for these test objects.

**Results.** In order to analyze sensitivity of our models to noisy inputs we reconstructed held-out test instances using our models given just ground truth bounding boxes. We compare various versions of our method using ground truth(Mask)/imperfect segmentations(SDS) and keypoints(KP)/our pose predictor(PP) for viewpoint estimation respectively. For pose prediction, we use the CNN-based system described in [17]. To obtain an approximate segmentation from the bounding box, we use the refinement stage of the state-of-the-art joint detection and segmentation system proposed in [42].

Table 2.2 shows that our results degrade gracefully from the fully annotated to the fully automatic setting. Our method is robust to some mis-segmentation owing to our shape model that prevents shapes from bending unnaturally to explain noisy silhouettes. Our reconstructions degrade slightly with imperfect pose initializations even though our projection parameter optimization deals with it to some extent. With predicted poses, we observe that sometimes even when our reconstructions look plausible, the errors can be high as the metrics are sensitive to bad alignment. The data sparsity issue is especially visible in the case of sofas where in a train/test

setting in Table 2.2 the numbers drop significantly with less training data (only 34 instances). Note we do not evaluate our bottom-up component as the PASCAL 3D+ meshes provided do not share the same high frequency shape details as the instance.

### 2.3.3 Fully Automatic Reconstruction

We qualitatively demonstrate reconstructions on automatically detected and segmented instances with 0.5 IoU overlap with the ground truth in whole images in PASCAL VOC using [42] in Figure 2.6. We can see that our method is able to deal with some degree of mis-segmentation. Some of our major failure modes include not being able to capture the correct scale and pose of the object and thus badly fitting to the silhouette in some cases.

## 2.4 Discussion

We proposed what may be the first approach to perform fully automatic object reconstruction from a single image on a large and realistic dataset. Critically, our deformable 3D shape model can be bootstrapped from easily acquired ground-truth 2D annotations thereby bypassing the need for a-priori manual mesh design or 3D scanning and making it possible for convenient use of these types of models on large real-world datasets (e.g. PASCAL VOC). We report an extensive evaluation of the quality of the learned 3D models on a 3D benchmarking dataset for PASCAL VOC [61] showing competitive results with models that specialize in shape reconstruction using ground truth annotations as inputs while demonstrating that our method is equally capable in the wild, on top of automatic object detectors.

Much research lies ahead, both in terms of improving the quality and the robustness of reconstruction at test time (both bottom-up and top-down components), developing benchmarks for joint recognition and reconstruction and relaxing the need for annotations during training: all of these constitute interesting and important directions for future work. More expressive non-linear shape models [63] may prove helpful (we present an instance in Chapter 4 with LSMs), as well as a tighter integration between segmentation and reconstruction.

# Chapter 3

# Amodal Completion and Size Constancy

In Chapter 2, we tackled the problem of learning object shape distributions and automatically localizing and reconstructing them in scenes. However, these object reconstructions are independent of each other, i.e. the relative sizes and depths of the objects are not taken into acount at all. In this chapter, we present ideas which allow us to take our object reconstructions from the previous chapter and assemble a coherent scene with them.

Consider Figure 3.1. Humans can effortlessly perceive two chairs of roughly the same height and tell that one is much closer than the other, though still further away than the person, who is taller than the chairs. Compare this to what a state-of-the-art object detector tells us about the image: that there are two chairs, 120 and 40 pixels tall, and one person with 200 pixels from top to bottom. How can we enable computer vision systems to move beyond this crude 2D representation and allow them to capture richer models of their environments, such as those that humans take for granted?

The 3D world is a lot more structured than it looks like from the retina (or from a camera sensor), where objects jump around with each saccade and grow and shrink as we move closer or farther from them. We do not perceive any of this because our brains have learned priors about how visual inputs correlate with the underlying environment, and this allows us to directly access realistic and rich models of scenes. The priors we use can be categorized as being related to either *geometry* or *familiarity*.

Image projection properties, such as the fact that the distance of an object from the camera dictates apparent size and that parallel lines in the scene vanish in the image, provide useful signal for perceiving structure. Familiarity cues are complemen-

This chapter is based on joint work with Shubham Tulsiani, João Carreira and Jitendra Malik [64], presented primarily as it appears in the ICCV 2015 proceedings. Statements throughout the chapter (such as references to "prior work") should be read with this context in mind.

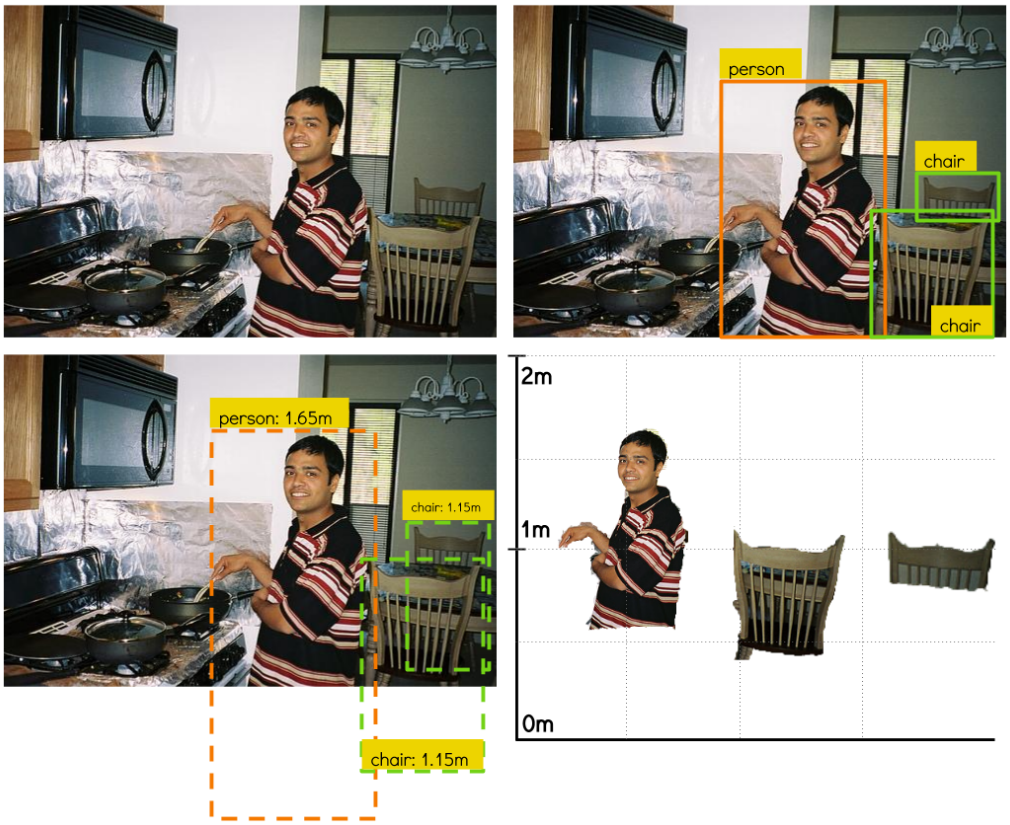


**Figure 3.1:** Perceiving the veridical size of objects in realistic scenes, from a single image, requires disentangling size and depth, being able to compensate for occlusions and to determine intrinsic camera parameters. We tackle all three of these problems, leveraging recent developments in object recognition and large annotated object and scene datasets.

tary and impose expectations on individual objects and configurations – we expect most objects to be supported by another surface and we have the notion of *familiar size* – similar objects are of similar sizes. In this chapter, we exploit geometry and familiarity cues and develop a framework to build richer models of the visual input than those given by current computer vision systems, which are still largely confined to the 2D image plane.

The notion that certain geometrical cues can aid perception has been known since the time of Euclid - the points in the image where objects touch the ground together with their perceived heights allows inference of real world object size ratios [65]. Familiarity cues, on the other hand must be learned, which can be done using available annotations and building upon rapid recent progress in object recognition, more robustly harnessed to explain novel images. Similar ideas have been proposed by Hoiem *et al.* [66, 67] and Gupta *et al.* [21] who studied the interaction between object detection and scene layout estimation and showed that, by reasoning over object sizes within their 3D environment, as opposed to within the image, one could perform bet-

ter object detection. Lalonde *et al.* [68] and Russell *et al.* [69] also tackled a problem similar to operationalizing size constancy and inferred object sizes of annotated objects. These works, while sharing similar goals to ours, were limited in their scope as they assumed fully visible instances - object recognition technology at the time being a limiting factor. In this chapter, we aim for veridical size estimation in more realistic settings – where occlusions are the rule rather than the exception. Occlusions present a significant technical challenge as they break down a number of assumptions(*e.g.* in Figure 3.1 not modeling occlusions would yield an incorrect estimate of the relative depths of the two chairs shown).

To overcome these challenges, we first introduce amodal completion. This is a very well studied ability of human perception, primarily in the context of amodal edge perception [70], building on theories of *good continuation* [71]. In the context of objects, amodal completion manifests itself as inference of the complete shape of the object despite visual evidence for only parts of it [72]. In Section 3.1, we tackle the amodal completion task and frame it as a recognition problem, formalized as predicting the full extent of object bounding boxes in an image, as opposed to only the visible extent. We build amodal extent predictors based on convolutional neural networks which we train on the challenging PASCAL VOC dataset. In Section 3.2, we propose a formulation that, leveraging amodally completed objects, can disentangle relative object sizes and object distances to the camera. This geometric reasoning allows us only to infer distances for objects up to a scaling ambiguity in each image. To overcome this ambiguity, we show in Section 3.3 that it is possible to leverage statistical dependencies between scenes and intrinsic camera parameters, and learn to predict focal lengths of scenes from large scale scene datasets. Finally, we present qualitative results exhibiting veridical size estimation in complex scenes.

## 3.1 Amodal Completion

> "Almost *nothing* is visible in its entirety, yet almost *everything* is perceived as a whole and complete"
>
> *Stephen Palmer*

Classic computer vision approaches have traditionally been impoverished by trying to explain just what we see in an image. For years, standard benchmarks have focused on explaining the visible evidence in the image - not the world behind it. For example, the well-studied task of predicting the bounding box around the visible pixels of an object has been the goal of current object detection systems. As humans, not only can we perceive the visible parts of the chair depicted in Figure 3.1, we can confidently infer the full extent of the actual chair.

This representation of objects, that humans can effortlessly perceive, is significantly richer than what current systems are capable of inferring. We take a step

forward towards achieving similar levels of understanding by attacking the task of perceiving the actual extent of the object, which we denote as *amodal completion*. The amodal representation of objects enables us to leverage additional scene information such as support relationships, occlusion orderings *etc.* For example, given the amodal and visible extents of two neighboring objects in the image, one can figure out if one is occluded by the other. Explicitly modeling amodal representations also allow us to implicitly model occlusions patterns rather than trying to "explain them away" while detecting objects. As described in Section 3.2, we can use these representations to infer real world object sizes and their relative depths just from images.

The primary focus of object recognition systems [15, 73] has been to localize and identify objects, despite occlusions, which are usually handled as noise. Several recently proposed recognition systems do explicitly model occlusion patterns along with detections and provide a mechanism for obtaining amodal extent of the object [74, 75, 76]. However, these approaches have been shown to work only on specific categories and rely on available shape models or depth inputs, for learning to reason over occlusions. In contrast, we aim to provide a generic framework that is not limited by these restrictions. Our proposed framework is described below.

### 3.1.1 Learning to Predict Amodal Boxes

Given a candidate visible bounding box, we tackle the task of amodal completion – the input to our system is some modal bounding box (*e.g.* obtained via a detection system) and we aim to predict the amodal extent for the object. We frame this task as predicting the amodal bounding box, which is defined as the bounding box of an object in the image plane if the object were completely visible, *i.e.* if inter-object occlusions and truncations were absent. The problem of amodal box prediction can naturally be formulated as a regression task - given a noisy modal bounding box of an object we regress to its amodal bounding box coordinates. The amodal prediction system is implicitly tasked with learning common occlusion/truncation patterns and their effects on visible object size. It can subsequently infer the correct amodal coordinates using the previously learned underlying visual structure corresponding to occlusion patterns. For example, the learner can figure out that chairs are normally vertically occluded by tables and that it should extend the bounding box vertically to predict the full extent of the chair.

Let $b = (x, y, w, h)$ be a candidate visible (or modal) bounding box our amodal prediction system receives ($(x, y)$ are the co-ordinates of the top-left corner and $(w, h)$ are the width and height of the box respectively) and $b^* = (x^*, y^*, w^*, h^*)$ be the amodal bounding box of the corresponding object, our regression targets are $(\frac{x-x^*}{w}, \frac{y-y^*}{h}, \frac{(x+w)-(x^*+w^*)}{w}, \frac{h-h^*}{h})$. Our choice of targets is inspired by the fact that for the $y$ dimension, the height and bottom of the box are the parameters we actually care about (see Section 3.2) whereas along the $x$ dimension the left co-ordinate is not necessarily more important than the right.

**Learning:** We use a Convolutional Neural Network (CNN) [77, 14] based framework to predict the co-ordinates of the amodal bounding box. The hypothesis is that the amodal prediction task can be reliably addressed given just the image corresponding to the visible object region – seeing the left of a car is sufficient to unambiguously infer the full extent without significantly leveraging context. Based on this observation, we extract from input image $I$, the region corresponding to the detection box $b$ and train the CNN using targets derived as above from the amodal box $b^*$. We impose an $L_2$ penalty on the targets and regress from the extracted CNN image features to the targets. We initialize our model using the AlexNet [78] CNN pretrained for Imagenet [79] classification and then finetune the model specific to our task using backpropagation. Training is carried out with jittered instances of the ground truth bounding box to enable generalization from noisy settings such as detection boxes and also serve as data augmentation.

We train two variants of the above network - class-specific and class agnostic. Both these systems comprise of 5 convolutional layers followed by 3 fully-connected layers. The class-specific network has separate outputs in the last layers for different classes and is trained with positive examples from a specific class whereas the class agnostic network has a single set of outputs across all classes. Intuitively, the class-specific network learns to leverage occlusion patterns specific to a particular class (*e.g.* chair occluded by a table) whereas the class agnostic network tries to learn occlusion patterns common across classes. Another argument for a class agnostic approach is that it is unreasonable to expect annotated amodal bounding box data for a large number of categories. A two-stage system, where we first predict the visible bounding box candidates and then regress from them to amodal boxes, enables leveraging these class agnostic systems to generalize to more categories. As we demonstrate in Section 3.2, this class agnostic network can be applied to novel object categories to learn object sizes.

### 3.1.2 Quantitative Evaluation

**Dataset:** For the purpose of amodal bounding box prediction, we need annotations for amodal bounding boxes (unlike visible bounding box annotations present in all standard detection datasets). We use the PASCAL 3D+ [61] dataset which has approximate 3D models aligned to 12 rigid categories on PASCAL VOC [43] to generate these amodal bounding box annotations. It also contains additional annotations for images from ImageNet [79] for each of these categories (22k instances in total from ImageNet). For example, it has between 4 different models aligned to "chair" and 10 aligned to "cars". The different models primarily distinguish between subcategories (but might also be redundant). The 3D models in the dataset are first aligned coarsely to the object instances and then further refined using keypoint annotations. As a consequence, they correctly capture the amodal extent of the object and allow us to obtain amodal ground-truth. We project the 3D model fitted per instance into

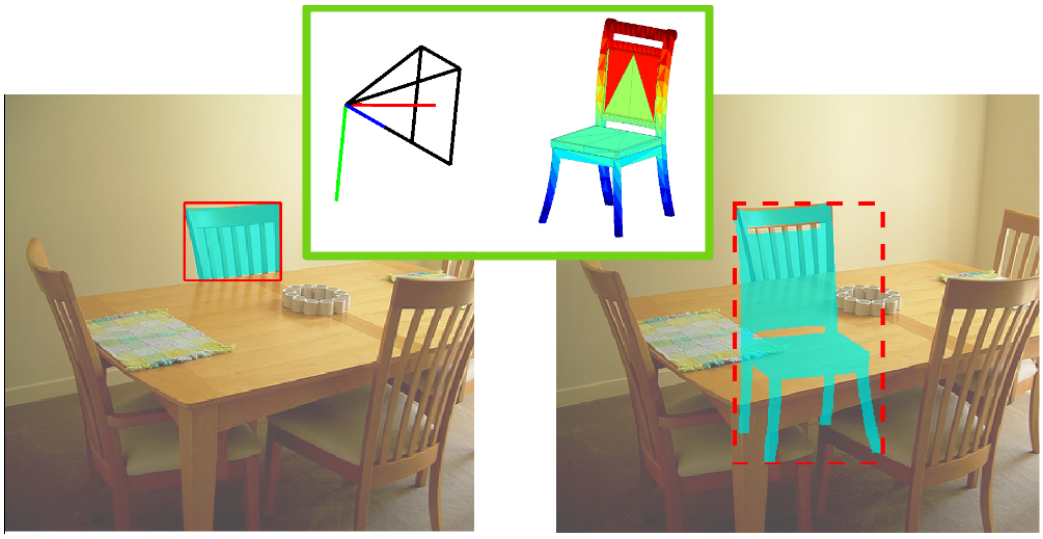

**Figure 3.2:** Generating amodal bounding boxes for instances in PASCAL VOC. We use the 3D models aligned to images from the PASCAL 3D+ [61] and render them with their annotated 3D pose to obtain binary masks. We then use the tightest fitting bounding box around the mask as our ground truth amodal bounding box.

the image, extract the binary mask of the projection and fit a tight bounding box around it which we treat as our amodal box (Figure 3.2). We train our amodal box regressors on the detection training set of PASCAL VOC 2012 (*det-train*) and the additional images from ImageNet for these 12 categories which have 3D models aligned in PASCAL 3D+ and test on the detection validation set (*det-val*) from the PASCAL VOC 2012 dataset.

**Experiments:** We benchmark our amodal bounding box predictor under two settings - going from ground truth visible bounding boxes to amodal boxes and in a detection setting where we predict amodal bounding boxes from noisy detection boxes. We compare against the baseline of using the modal bounding box itself as the amodal bounding box (*modal bbox*) which is in fact the correct prediction for all untruncated instances. Table 3.1 summarizes our experiments in the former setting where we predict amodal boxes from visible ground truth boxes on various subsets of the dataset and report the mean IoU of our predicted amodal boxes with the ground truth amodal boxes generated from PASCAL 3D+. As expected, we obtain the greatest boost over the baseline for truncated instances. Interestingly, the class agnostic network performs as well the class specific one signaling that occlusion patterns span across classes and one can leverage these similarities to train a generic amodal box regressor.

To test our amodal box predictor in a noisy setting, we apply it on bounding boxes predicted by the RCNN[15] system from Girshick *et al.*. We assume a detection be

correct if the RCNN bounding box has an IoU $> 0.5$ with the ground truth visible box *and* the predicted amodal bounding box also has an IoU $> 0.5$ with the ground truth amodal box. We calculate the average precision for each class under the above definition of a "correct" detection and call it the Amodal $AP$ (or $AP^{am}$). Table 3.2 presents our $AP^{am}$ results on VOC2012 *det-val*. As we can see again, the class agnostic and class specific systems perform very similarly. The notable improvement is only in a few classes (*e.g.* diningtable and boat) where truncated/occluded instances dominate. Note that we do not rescore the RCNN detections using our amodal predictor and thus our performance is bounded by the detector performance. Moreover, the instances detected correctly by the detector tend to be cleaner ones and thus the baseline (*modal bbox*) of using the detector box output as the amodal box also does reasonably well. Our RCNN detector is based on the VGG16 [80] architecture and has a mean $AP$ of 57.0 on the 12 rigid categories we consider.

| | **all** | **trunc/occ** | **trunc** | **occ** |
|---|---|---|---|---|
| **modal bbox** | 0.66 | 0.59 | 0.52 | 0.64 |
| **class specific** | 0.68 | 0.62 | 0.57 | 0.65 |
| **class agnostic** | 0.68 | 0.62 | 0.56 | 0.65 |

**Table 3.1:** Mean IoU of amodal boxes predicted from the visible bounding box on various subsets of the validation set in PASCAL VOC. Here *occ* and *trunc* refer to occluded and truncated instances respectively. The class specific and class agnostic methods refer to our variations of the training the amodal box regressors (see text for details) and modal bbox refers to the baseline of using the visible/modal bounding box itself as the predicted amodal bounding box.

| | **aero** | **bike** | **boat** | **bottle** | **bus** | **car** | **chair** | **table** | **mbike** | **sofa** | **train** | **tv** | **mean** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **modal bbox** | 70.0 | 66.2 | 23.9 | 35.1 | 76.4 | 57.7 | 28.9 | 24.2 | 68.3 | 45.8 | 58.1 | 59.6 | 51.2 |
| **class specific** | 69.5 | 67.2 | **26.9** | 36.0 | **77.0** | **61.4** | **31.4** | 29.2 | **69.0** | **49.4** | **59.3** | 59.5 | **53.0** |
| **class agnostic** | **70.0** | **67.5** | 26.8 | **36.3** | 76.8 | 61.3 | 31.1 | **30.9** | 68.9 | 48.4 | 58.6 | **59.6** | **53.0** |

**Table 3.2:** $AP^{am}$ for our amodal bounding box predictors on VOC 2012 *det-val*. $AP^{am}$ is defined as the average precision when a detection is assumed to be correct only when both the modal and amodal bounding boxes have IoU $> 0.5$ with their corresponding ground truths.

Armed with amodal bounding boxes, we now show how we tackle the problem of inferring real world object sizes from images.

## 3.2 Disentangling Size and Depth

Monocular cues for depth perception have been well-studied in psychology literature and there are two very important cues which emerge that tie object size and

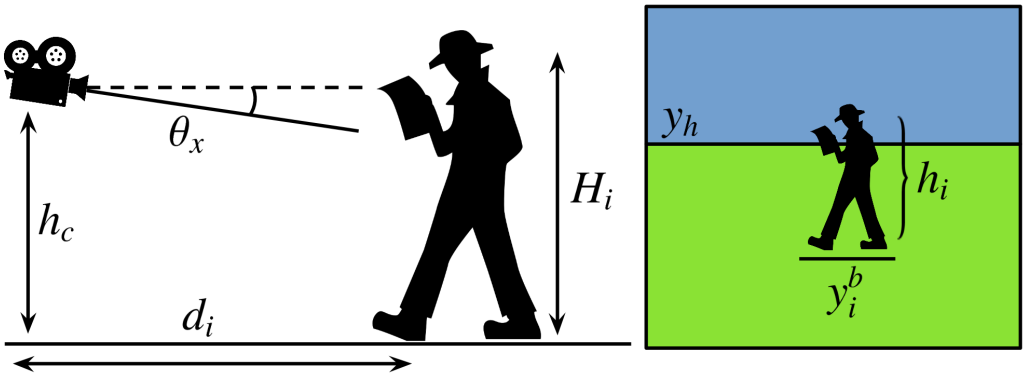


**Figure 3.3:** Toy example illustrating our camera model and parameters. Please refer to the text for detailed explanations.

depth - namely familiar size and relative size. Familiar size is governed by the fact that the visual angle subtended by an object decreases with distance from the observer and prior knowledge about the actual size of the object can be leveraged to obtain absolute depth of the object in the scene. Relative size, on the other hand, helps in explaining relative depths and sizes of objects - if we know that two objects are of similar sizes in the real world, the smaller object in the image appears farther. Another simple cue for depth perception arises due to perspective projection - an object further in the world appears higher on the image plane. Leveraging these three cues, we show that one can estimate real world object sizes from just images. In addition to object sizes, we also estimate a coarse viewpoint for each image in the form of the horizon and camera height.

The main idea behind the algorithm is to exploit pairwise size relationships between instances of different object classes in images. As we will show below, given support points of objects on the ground and some rough estimate of object sizes, one can estimate the camera height and horizon position in the image - and as a result relative object depths. And in turn, given object heights in the image and relative depths, one can figure out the real world object scale ratios. Finally, exploiting these pairwise size evidences across images, we solve for absolute real world sizes (upto a common scale factor or the metric scale factor). Note that we use size and height interchangeably here as our notion of object size here actually refers to the object height.

### 3.2.1 Camera Model

We use a simplified perspective camera model similar to Hoiem *et al.* [66]. Let $f$ be the focal length of the camera, $\theta_x$ the camera tilt angle along the x-axis, $h_c$ the height of the camera, $y_h$ be the horizon position in the image, $y_i^b$ be the ground support point for the $i^{th}$ object in the image and $d_i$ be the distance of the $i^{th}$ object

from the camera along the camera axis ($z$ axis). We assume that the images have been corrected for camera roll and all pixel co-ordinates are with respect to the optical center (assumed to be center of the image). Figure 3.3 provides a toy illustration of our model and parameters.

Assuming that the world frame is centered at the camera with its $y$ axis aligned with the ground, the projection of a world point $\mathbf{X} = (X_w, Y_w, Z_w)$ in the image in homogeneous co-ordinates is given by:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{Z_w} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x & 0 \\ 0 & -\sin\theta_x & \cos\theta_x & 0 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

For a world point corresponding to the ground contact point of object $i$, given by $(X_w, -h, d_i)$, its corresponding $y$ co-ordinate in the image $y_i^b$ is given by: $y_i^b = f\frac{(-h_c/d_i + \tan\theta_x)}{1+(h_c/d_i)\tan\theta_x}$ Under the assumption of the tilt angle being small ($\tan\theta_x \approx \theta_x$) and height of the camera being not too large compared to object distance ($h\theta_x \ll d_i$), our approximation is

$$y_i^b = -\frac{fh_c}{d_i} + f\theta_x \tag{3.1}$$

Here $f\theta_x$ corresponds to the position of the horizon ($y_h$) in the image. Repeating the above calculation for the topmost point of the object and subtracting from Eq. 3.1, we obtain

$$h_i = \frac{fH_i}{d_i} \tag{3.2}$$

where $h_i$ refers to the height of the object in the image and $H_i$ is the real world height of the object. Our model makes some simplifying assumptions about the scene namely, objects are assumed to rest on the same horizontal surface (here, the ground) and camera tilt is assumed to be small. We observe that for the purpose of size inference, these assumptions turn out to be reasonable and allow us to estimate heights of objects fairly robustly.

### 3.2.2 Inferring Object Sizes

The important observation here is that the sizes of objects in an object category are not completely random - they potentially follow a multimodal distribution. For example, different subcategories of boats may represent the different modes of the size distribution. Given some initial sizes and size cluster estimates, our algorithm for size estimation (Algorithm 1) works by estimating the horizon and camera height per image (by solving a least squares problem using Eq. 3.1 and Eq. 3.2 for all the objects in an image). With the horizon and height estimated per image, we obtain

**Algorithm 1** Object Size Estimation

**Initialize:**
Initial size estimates $\mathbf{H}$ and cluster assignments
**while** not converged **do**
  **for all** images $k \in$ Dataset **do**
    $(h_c, y_h) \leftarrow \text{SolveLeastSquares}(y_b, h, \mathbf{H})$
    **for all** pairs $(i, j)$ of objects in $k$ **do**
      $\frac{H_i}{H_j} \leftarrow \frac{h_i}{h_j} \frac{y_j^b - y_h}{y_i^b - y_h}$ ▷ (1)
    **end for**
  **end for**
  $\log \mathbf{H} \leftarrow$ least squares with pairwise constraints (1)
  GMM cluster log scales ($\log \mathbf{H}$)
  Reassign objects to clusters
**end while**

pairwise height ratios $\frac{H_i}{H_j} = \frac{h_i}{h_j} \frac{y_j^b - y_h}{y_i^b - y_h}$ for each pair of objects in an image. We obtain multiple such hypotheses across the dataset which we use to solve a least squares problem for $\log \mathbf{H}$ - the log height for each size cluster. Finally, we cluster the log sizes obtained in the previous step to obtain new size clusters and iterate. Note that $\mathbf{H}$ refers to the vector with heights of various classes and $H_i$ refers to the real world size of the $i^{th}$ object.

This particular model is equivalent to solving a latent variable model where the latent variables are the cluster memberships of the instances, the estimated variables are heights corresponding to the size clusters and the horizon and camera height for each image. The loss function we try to minimize is the mean squared error between the ground contact point predicted by the model and the amodal bounding box. Finally, the log of the object heights are assumed to be a Gaussian mixture. This final assumption ties in elegantly with psychophysics studies which have found that our mental representation of object size (referred to as assumed size [81, 82, 83]) is proportional to the logarithm of the real world object size [84].

Our image evidences in the above procedure include the ground support points and heights for all the objects in the image. Note that amodal bounding boxes for objects provide us exactly this information. They account for occlusions and truncations and give us an estimate of the full extent of the object in the image. The above algorithm with occluded/truncated visible bounding boxes would fail miserably and we use our amodal bounding box predictor to first "complete" the bounding boxes for us before using our size inference algorithm to infer object heights.

**Inferring Object Size Statistics on PASCAL VOC:** We used our size estimation system on PASCAL VOC to estimate size distributions of objects. First, we use

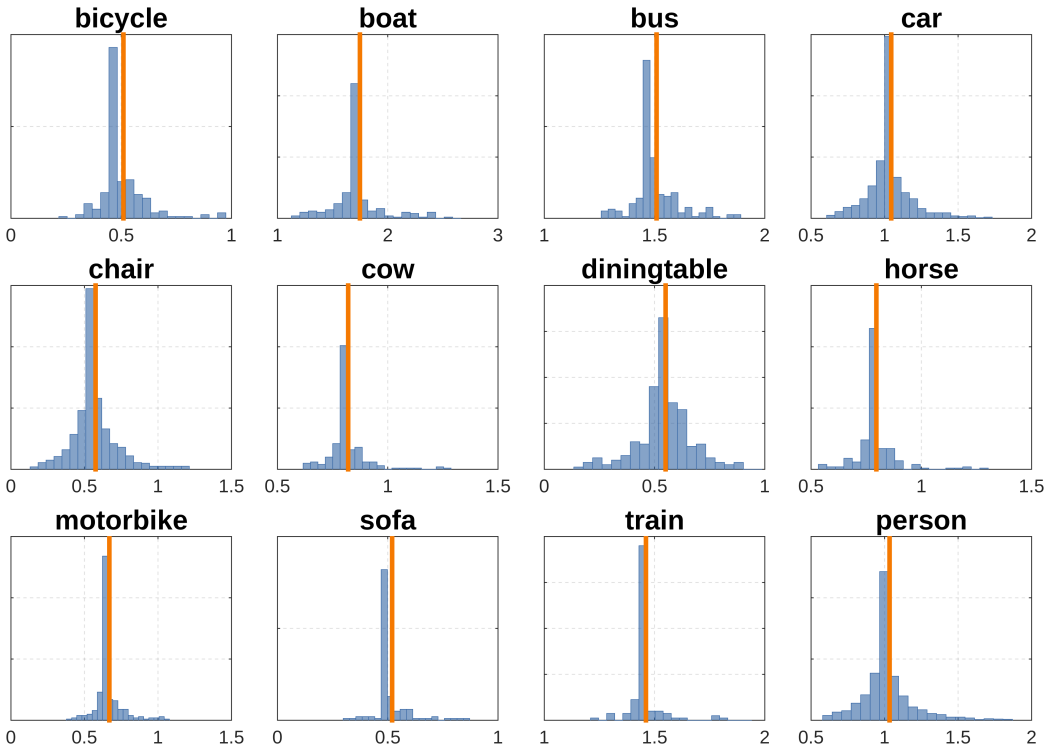


**Figure 3.4:** Inferred log size distributions of 12 object categories on PASCAL VOC. We use our class agnostic amodal bounding box predictor to predict amodal boxes for all instances in VOC 2012 *det-val* and use them with our object size estimation system to estimate size distributions for various categories. The plots above show distributions of the log size with the mean size being shown by the orange line.

our class agnostic amodal bounding box predictor on ground truth visible bounding boxes of all instances on VOC 2012 *det-val* to "upgrade" them to amodal boxes. We initialize our system with a rough mean height for each object class obtained from internet sources (Wikipedia, databases of cars *etc.*) and run our size estimation algorithm on these predicted amodal boxes. Figure 3.4 shows the distributions of log sizes of objects of various categories in PASCAL VOC. Most categories exhibit peaky distributions with classes such as "boat" and "chair" having longer tails owing to comparatively large intra class variation. Note that we experimented with using multiple size clusters per class for this experiment but the peaky, long tailed nature of these distributions meant that a single Gaussian capturing the log size distributions sufficed. In addition to inferring object sizes, we also infer the horizon position and height of the camera. The median height of the camera across the dataset was 1.4 metres (roughly the height at which people take images) and also exhibited a long tailed distribution (Figure 3.6). Some examples of amodal bounding boxes estimated for all instances from visible bounding boxes and horizons are shown in Figure 3.8.

## 3.3 Scenes and Focal Lengths

The focal length of a camera defines its field of view and hence determines how much of a scene is captured in an image taken by the camera. It is an important calibration parameter for obtaining metric, as opposed to projective, measurements from images. It is usually calibrated using multiple images of a known object [85], such as a chessboard, or as part of bundle adjustment [86], from multiple images of realistic scenes. Well known existing approaches require a minimum set of vanishing lines [87] or exploit Manhattan-world assumptions [88]. These techniques are very precise and elegant, but not generally applicable (e.g. beach or forest images, *etc.*).

We propose instead a learning approach that predicts focal length based on statistical dependencies between scene classes and fields of view. Given the same scene, images taken with large focal lengths will have fewer things in them than those captured with small focal lengths and this provides a cue for determining focal length. However certain scenes also have more things than others. This ambiguity can be resolved by training a predictor with many images of each scene class, taken with different focal lengths.

Additionally, certain scenes tend to be pictured with preferred focal lengths. As an example, consider a scene class of "pulpits". If a picture of a pulpit is taken with a short focal length, then the whole church will be visible and that image will not be tagged as a pulpit scene. In order for a pulpit to be dominant in a picture taken with a short focal length camera, then the photographer would have to be unnaturally close to it.

**Data:** We use the Places database [89], a large dataset that provides a dense sampling of scenes in natural images: it has 205 scene classes, as diverse as *swimming pool* and *rope bridge*, and 2.5 million images. We were able to scrape focal length metadata from EXIF tags of approximately 20k examples, on average 100 per class, and split these into a training set having 15k and a validation set of 5k images.

**Learning:** We considered the problem of predicting the ratio of the focal length to the camera sensor width, which when multiplied by the size of the image in pixels gives the desired the focal length in pixels. We clustered the logarithm of this ratio into 10 bins using k-means and formulated the prediction problem as classification, using a softmax loss. Images in the bin with highest and smallest focal length ratio are shown in Figure 3.5. We experimented finetuning different popular convolutional networks, including two trained on Imagenet classification – AlexNet [78] and VGG-Deep16 [80] – and a network trained on the Places scenes – the PlacesNet [89].

**Results:** The results are shown in Table 3.3 and suggest that focal length can indeed be predicted directly from images, at least approximately, and that pretraining

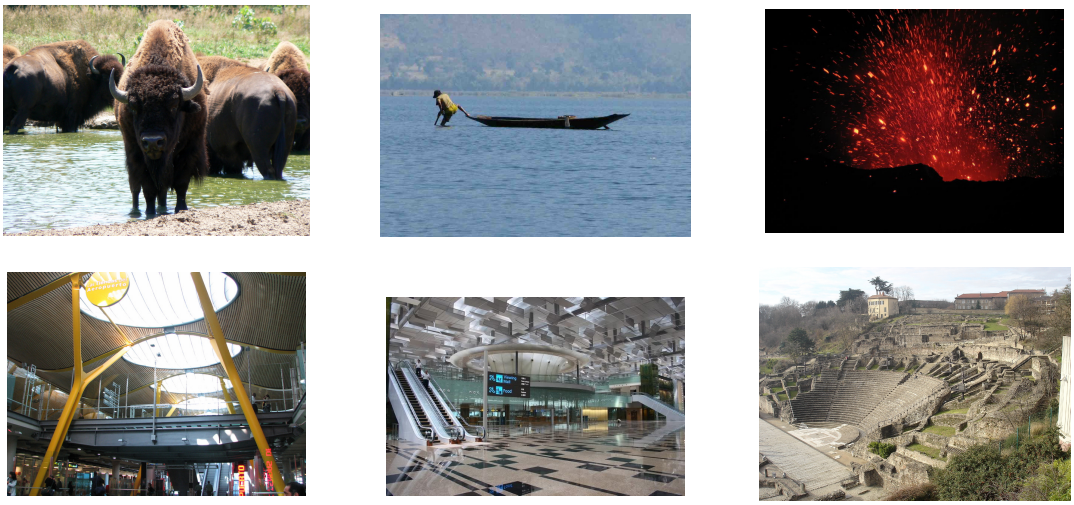

**Figure 3.5:** Example images from the Places dataset from clusters with the largest (up) and smallest (down) focal lengths. Note how images with small focal lengths tend to be more cluttered. A pattern we observed is that dangerous or unaccessible scenes, such as those having volcanos, wild animals and boats tend to be captured using very-high focal lengths, which is rational.

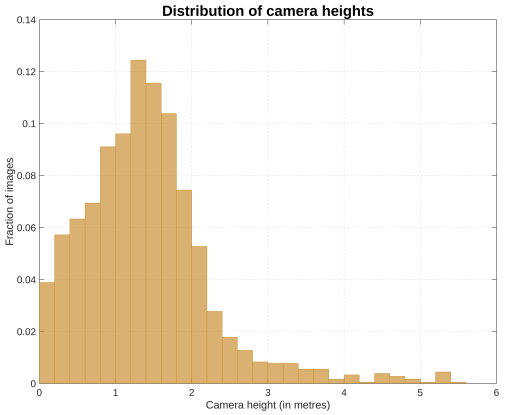


**Figure 3.6:** Distribution of camera heights as inferred on PASCAL VOC. It can be seen that the distribution is peaked around the height at which humans normally take pictures (1.4m) with a long tail.

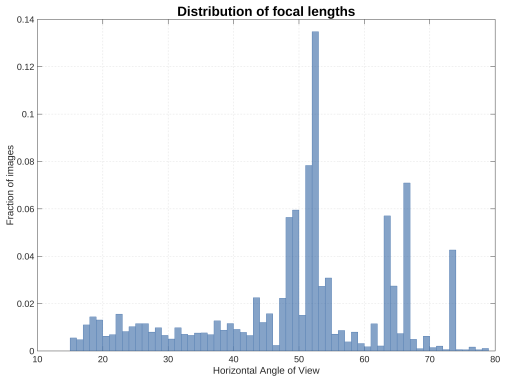


**Figure 3.7:** Distribution of camera focal lengths in the Places dataset shown as horizontal angle of view. The peaks in the distribution correspond to canonical focal lengths in popular wide angle and telephoto lenses. It can be seen that the images cover the full spectrum from wide angle views of scenes (corresponding to the right end of the plot) to close-ups (left end of the histogram).

on annotated scene class data makes a good match with this task. Our best model can predict correct focal length quite repeatably among the top-three and top-five predictions. As baselines, we measure chance performance, and performance when picking the mode of the distribution on the training set – the bin having most elements. The

| Method | top-1 | top-3 | top-5 |
|---|---|---|---|
| Chance | 90.0 | 70.0 | 50.0 |
| Mode Selection | 60.2 | 26.4 | 8.7 |
| AlexNet-Imagenet | 57.1 | 18.8 | 3.9 |
| VGG-Deep16-Imagenet | 55.8 | 15.9 | 3.3 |
| PlacesNet-Places | **54.3** | **15.3** | **3.1** |

**Table 3.3:** Focal length misclassification rate (top-1, top-3 and top-5 predictions) of networks pretrained on object images from Imagenet and the Places dataset. Lower is better.

(unequal) distribution of the focal lengths can be seen in Figure 3.7.

Note that our goal is not high precision of the type that is necessary for high-fidelity reconstruction; we aim for a coarse estimate of the focal length that can be robustly computed from natural images. Our results in this section are a first demonstration that this may be feasible.

## 3.4 Discussion

We have studied the problem of veridical size estimation in complex natural scenes, with the goal of enriching the visual representations inferred by current recognition systems. We presented techniques for performing amodal completion of detected object bounding boxes, which together with geometric cues allow us to recover relative object sizes, and hence achieve a desirable property of any perceptual system - size constancy. We have also introduced and demonstrated a learning-based approach for predicting focal lengths, which can allow for metrically accurate predictions when standard auto-calibration cues or camera metadata are unavailable. We strived for generality by leveraging recognition. This is unavoidable because the size constancy problem is fundamentally ill-posed and can only be dealt with probabilistically.

We also note that while the focus of our work is to enable veridical size prediction in natural scenes, the three components we have introduced to achieve this goal - amodal completion, geometric reasoning with size constancy and focal length prediction are generic and widely applicable. We provided individual evaluations of each of these components, which together with our qualitative results demonstrate the suitability of our techniques towards understanding real world images at a rich and general level, beyond the 2D image plane.

**Figure 3.8:** Amodal bounding box prediction and size estimation results on images in PASCAL VOC. The solid rectangles represent the visible bounding boxes and the dotted lines are the predicted amodal bounding boxes with heights in meters. The horizontal red line denotes the estimated horizon position for the image.

# Chapter 4

# Towards Unification: Learnt Stereo Machines

In Chapters 2 and 3, we presented algorithms which allow us to learn deformable 3D models of objects and subsequently learn statistics about their relative sizes and depths to assemble a coherent scene. In both cases, we focussed on making inferences on a single image. Moreover, we used recognition systems primarily as auxilliary systems to provide us cues about shape (e.g. instance segmentation, pose estimation in Chapter 2 and amodal bounding box prediction in 3). In this chapter, we revisit the problem of modelling 3D shapes from image collections and model them implicitly using powerful recognition systems (deep neural networks) rather than simple linear models. Additionally, instead of restricting ourselves to single view geometry prediction, we propose integrating information from multiple views in a geometrically consistent manner right within a convolutional neural network. Not only does this allow us to use multi-view cues while learning as in Chapter 2, but also enforce them during inference. We study these problems in the classical setting of multi-view stereo.

Multi-view stereopsis (MVS) is classically posed as the following problem - given a set of images with known camera poses, it produces a geometric representation of the underlying 3D world. This representation can be a set of disparity maps, a 3D volume in the form of voxel occupancies, signed distance field *etc.* An early example of such a system is the stereo machine from Kanade *et al.* [91] that computes disparity maps from images streams from six video cameras. Modern approaches focus on acquiring the full 3D geometry in the form of volumetric representations or polygonal meshes [6]. The underlying principle behind MVS is simple - a 3D point looks locally similar when projected to different viewpoints [92]. Thus, classical methods use the basic

This chapter is based on joint work with Christian Häne and Jitendra Malik [90], presented primarily as it appears in the NIPS 2017 proceedings. Statements throughout the chapter (such as references to "prior work") should be read with this context in mind.

principle of finding dense correspondences in images and triangulate to obtain a 3D reconstruction.

The question we try to address in this chapter is can we *learn* a multi-view stereo system? For the binocular case, Becker and Hinton [93] demonstrated that a neural network can learn to predict a depth map from random dot stereograms. A recent work [94] shows convincing results for binocular stereo by using an end-to-end learning approach with binocular geometry constraints.

In this work, we present Learnt Stereo Machines (LSM) - a system which is able to reconstruct object geometry as voxel occupancy grids or per-view depth maps from a small number of views, including just a single image. We design our system inspired by classical approaches while learning each component from data embedded in an end to end system. LSMs have built in projective geometry, enabling reasoning in metric 3D space and effectively exploiting the geometric structure of the MVS problem. Compared to classical approaches, which are designed to exploit a specific cue such as silhouettes or photo-consistency, our system learns to exploit the cues that are relevant to the particular instance while also using priors about shape to predict geometry for unseen regions.

Recent work from Choy *et al.* [95] (3D-R2N2) trains convolutional neural networks (CNNs) to predict object geometry given only images. While this work relied primarily on semantic cues for reconstruction, our formulation enables us to exploit strong geometric cues. In our experiments, we demonstrate that a straightforward way of incorporating camera poses for volumetric occupancy prediction does not lead to expected gains, while our geometrically grounded method is able to effectively utilize the additional information.

Classical multi-view stereopsis is traditionally able to handle both objects and scenes - we only showcase our system for the case of objects with scenes left for future work. We thoroughly evaluate our system on the synthetic ShapeNet [96] dataset. We compare to classical plane sweeping stereo, visual hulls and several challenging learning-based baselines. Our experiments show that we are able to reconstruct objects with fewer images than classical approaches. Compared to recent learning based reconstruction approaches, our system is able to better use camera pose information leading to significantly large improvements while adding more views. Finally, we show successful generalization to unseen object categories demonstrating that our network goes beyond semantic cues and strongly uses geometric information for unified single and multi-view 3D reconstruction.

## 4.1 Related Work

Extracting 3D information from images is one of the classical problems in computer vision. Early works focused on the problem of extracting a disparity map from a binocular image pair [97]. We refer the reader to [98] for an overview of classical

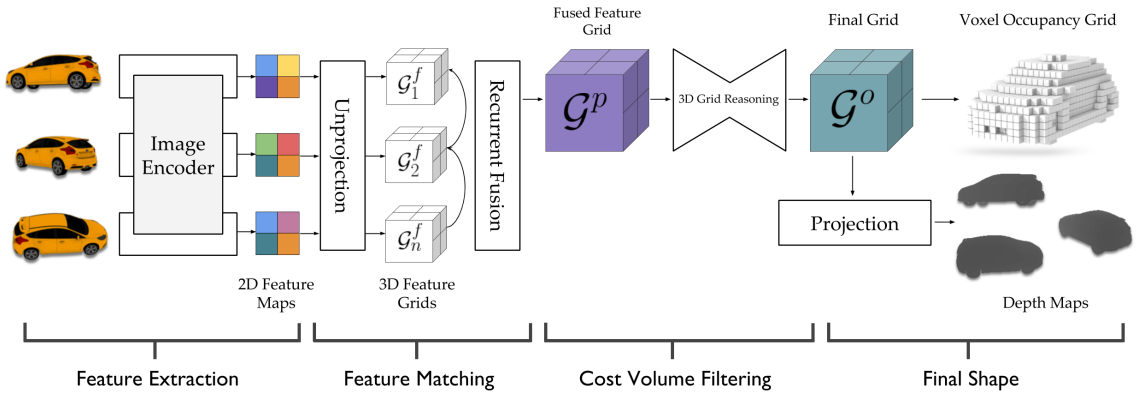


**Figure 4.1:** Overview of a Learnt Stereo Machine (LSM). It takes as input one or more views and camera poses. The images are processed through a feature encoder which are then unprojected into the 3D world frame using a differentiable unprojection operation. These grids $\{G_i^f\}_{i=1}^n$ are then matched in a recurrent manner to produce a fused grid $\mathcal{G}^p$ which is then transformed by a 3D CNN into $\mathcal{G}^o$. LSMs can produce two kinds of outputs - voxel occupancy grids (Voxel LSM) decoded from $\mathcal{G}^o$ or per-view depth maps (Depth LSM) decoded after a projection operation.

binocular stereo matching algorithms. In the multi-view setting, early work focused on using silhouette information via visual hulls [52], incorporating photo-consistency to deal with concavities (photo hull) [92], and shape refinement using optimization [99, 100, 101, 102]. [103, 104, 105] directly reason about viewing rays in a voxel grid, while [106] recovers a quasi dense point cloud. In our work, we aim to learn a multi-view stereo machine grounded in geometry, that learns to use these classical constraints while also being able to reason about semantic shape cues from the data. Another approach to MVS involves representing the reconstruction as a collection of depth maps [107, 108, 109, 110, 111]. This allows recovery of fine details for which a consistent global estimate may be hard to obtain. These depth maps can then be fused using a variety of different techniques [112, 113, 114, 115, 116]. Our learnt system is able to produce a set of per-view depth maps along with a globally consistent volumetric representation which allows us to preserve fine details while conforming to global structure.

Learning has been used for multi-view reconstruction in the form of shape priors for objects [10, 117, 118, 119, 120, 18], or semantic class specific surface priors for scenes [121, 122, 123]. These works use learnt shape models and either directly fit them to input images or utilize them in a joint representation that fuses semantic and geometric information. Most recently, CNN based learning methods have been proposed for 3D reconstruction by learning image patch similarity functions [124, 125, 126] and end-to-end disparity regression from stereo pairs [127, 94]. Approaches which predict shape from a single image have been proposed in form of direct depth map regression [128, 129, 37], generating multiple depth maps from novel viewpoints [130], producing voxel occupancies [95, 131], geometry images [132] and point clouds [133].

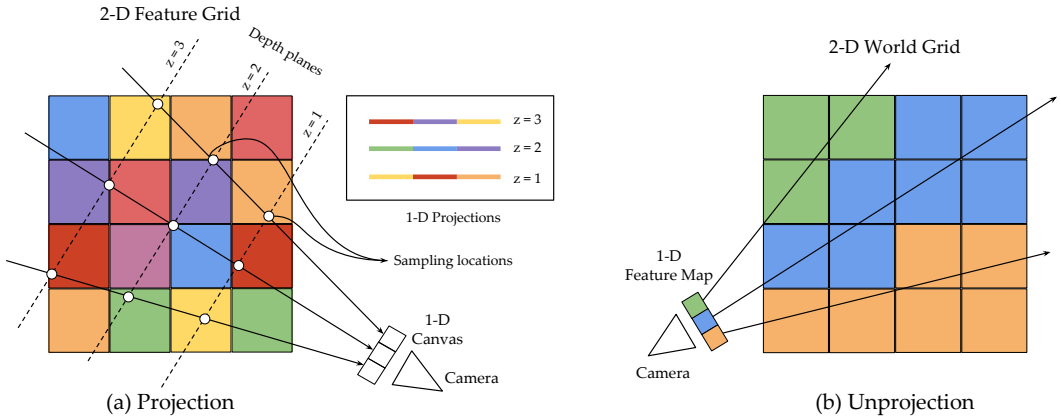


**Figure 4.2:** Illustrations of projection and unprojection operations between 1D maps and 2D grids. (a) The projection operation samples values along the ray at equally spaced z-values into a 1D canvas/image. The sampled features (shown by colors here) at the z planes are stacked into channels to form the projected feature map. (b) The unprojection operation takes features from a feature map (here in 1-D) and places them along rays at grid blocks where the respective rays intersect. Best viewed in color.

[134] study a related problem of view interpolation, where a rough depth estimate is obtained within the system.

A line of recent works, complementary to ours, has proposed to incorporate ideas from multi-view geometry in a learning framework to train single view prediction systems [135, 136, 137, 138, 139] using multiple views as supervisory signal. These works use the classical cues of photo-consistency and silhouette consistency only during training - their goal during inference is to only perform single image shape prediction. In contrast, we also use geometric constraints during inference to produce high quality outputs.

Closest to our work is the work of Kendall *et al.* [94] which demonstrates incorporating binocular stereo geometry into deep networks by formulating a cost volume in terms of disparities and regressing depth values using a differentiable arg-min operation. We generalize to multiple views by tracing rays through a discretized grid and handle variable number of views via incremental matching using recurrent units. We also propose a differentiable projection operation which aggregates features along viewing rays and learns a nonlinear combination function instead of using the differentiable arg-min which is susceptible to multiple modes. Moreover, we can also infer 3D from a single image during inference.

## 4.2 Learnt Stereo Machines

Our goal in this chapter is to design an end-to-end learnable system that produces a 3D reconstruction given one or more input images and their corresponding camera poses. To this end, we draw inspiration from classical geometric approaches where the underlying guiding principle is the following - the reconstructed 3D surface has to be photo-consistent with all the input images that depict this particular surface. Such approaches typically operate by first computing dense features for correspondence matching in image space. These features are then assembled into a large cost volume of geometrically feasible matches based on the camera pose. Finally, the optimum of this matching volume (along with certain priors) results in an estimate of the 3D volume/surface/disparity maps of the underlying shape from which the images were produced.

Our proposed system, shown in Figure 4.1, largely follows the principles mentioned above. It uses a discrete grid as internal representation of the 3D world and operates in metric 3D space. The input images $\{I_i\}_{i=1}^n$ are first processed through a shared image encoder which produces dense feature maps $\{\mathcal{F}_i\}_{i=1}^n$, one for each image. The features are then *unprojected* into 3D feature grids $\{\mathcal{G}_i^f\}_{i=1}^n$ by rasterizing the viewing rays with the known camera poses $\{\mathcal{P}_i\}_{i=1}^n$. This unprojection operation aligns the features along epipolar lines, enabling efficient local matching. This matching is modelled using a recurrent neural network which processes the unprojected grids sequentially to produce a grid of local matching costs $\mathcal{G}^p$. This cost volume is typically noisy and is smoothed in an energy optimization framework with a data term and smoothness term. We model this step by a feed forward 3D convolution-deconvolution CNN that transforms $\mathcal{G}^p$ into a 3D grid $\mathcal{G}^o$ of smoothed costs taking context into account. Based on the desired output, we propose to either let the final grid be a volumetric occupancy map or a grid of features which is *projected* back into 2D feature maps $\{\mathcal{O}_i\}_{i=1}^n$ using the given camera poses. These 2D maps are then mapped to a view specific representation of the shape such as a per view depth/disparity map. The key components of our system are the differentiable projection and unprojection operations which allow us to learn the system end-to-end while injecting the underlying 3D geometry in a metrically accurate manner. We refer to our system as a Learnt Stereo Machine (**LSM**). We present two variants - one that produces per voxel occupancy maps (**V**oxel **LSM**) and another that outputs a depth map per input image (**D**epth **LSM**) and provide details about the components and the rationale behind them below.

### 4.2.1 2D Image Encoder

The first step in a stereo algorithm is to compute a good set of features to match across images. Traditional stereo algorithms typically use raw patches as features. We model this as a feed forward CNN with a convolution-deconvolution architecture

with skip connections (UNet) [140] to enable the features to have a large enough receptive field while at the same time having access to lower level features (using skip connections) whenever needed. Given images $\{I_i\}_{i=1}^n$, the feature encoder produces dense feature maps $\{\mathcal{F}_i\}_{i=1}^n$ in 2D image space, which are passed to the unprojection module along with the camera parameters to be lifted into metric 3D space.

### 4.2.2 Differentiable Unprojection

The goal of the unprojection operation is to lift information from 2D image frame to the 3D world frame. Given a 2D point $p$, its feature representation $\mathcal{F}(p)$ and our global 3D grid representation, we replicate $\mathcal{F}(p)$ along the viewing ray for $p$ into locations along the viewing ray in the metric 3D grid (a 2D illustration is presented in Figure 4.2). In the case of perspective projection specified by an intrinsic camera matrix $K$ and an extrinsic camera matrix $[R|t]$, the unprojection operation uses this camera pose to trace viewing rays in the world and copy the image features into voxels in this 3D world grid. Instead of analytically tracing rays, given the centers of blocks in our 3D grid $\{X_w^k\}_{k=1}^{N_V}$, we compute the feature for $k^{th}$ block by projecting $\{X_w^k\}$ using the camera projection equations $p_k' = K[R|t]X_w^k$ into the image space. $p_k'$ is a continuous quantity whereas $\mathcal{F}$ is defined on at discrete 2D locations. Thus, we use the differentiable bilinear sampling operation to sample from the discrete grid [141] to obtain the feature at $X_w^k$.

Such an operation has the highly desirable property that features from pixels in multiple images that may correspond to the same 3D world point unproject to the same location in the 3D grid - trivially enforcing epipolar constraints. As a result, any further processing on these unprojected grids has easy access to corresponding features to make matching decisions foregoing the need for long range image connections for feature matching in image space. Also, by projecting discrete 3D points into 2D and bilinearly sampling from the feature map rather than analytically tracing rays in 3D, we implicitly handle the issue where the probability of a grid voxel being hit by a ray decreases with distance from the camera due to their projective nature. In our formulation, every voxel gets a "soft" feature assigned based on where it projects back in the image, making the feature grids $\mathcal{G}^f$ smooth and providing stable gradients. This geometric procedure of lifting features from 2D maps into 3D space is in contrast with recent learning based approaches [95, 130] which either reshape flattened feature maps into 3D grids for subsequent processing or inject pose into the system using fully connected layers. This procedure effectively saves the network from having to implicitly *learn* projective geometry and directly bakes this given fact into the system. In LSMs, we use this operation to unproject the feature maps $\{\mathcal{F}_i\}_{i=1}^n$ in image space produced by the feature encoder into feature grids $\{\mathcal{G}_i^f\}_{i=1}^n$ that lie in metric 3D space.

For single image prediction, LSMs cannot match features from multiple images to reason about where to place surfaces. Therefore, we append geometric features

along the rays during the projection and unprojection operation to facilitate single view prediction. Specifically, we add the depth value and the ray direction at each sampling point.

### 4.2.3 Recurrent Grid Fusion

The 3D feature grids $\{\mathcal{G}_i^f\}_{i=1}^n$ encode information about individual input images and need to be fused to produce a single grid so that further stages may reason jointly over all the images. For example, a simple strategy to fuse them would be to just use a point-wise function - *e.g.* max or average. This approach poses an issue where the combination is too spatially local and early fuses all the information from the individual grids. Another extreme is concatenating all the feature grids before further processing. The complexity of this approach scales linearly with the number of inputs and poses issues while processing a variable number of images. Instead, we choose to processed the grids in a sequential manner using a recurrent neural network. Specifically, we use a 3D convolutional variant of the Gated Recurrent Unit (GRU) [142, 143, 95] which combines the grids $\{\mathcal{G}_i^f\}_{i=1}^n$ using 3D convolutions (and non-linearities) into a single grid $\mathcal{G}^p$. Using convolutions helps us effectively exploit neighborhood information in 3D space for incrementally combining the grids while keeping the number of parameters low. Intuitively, this step can be thought of as mimicking incremental matching in MVS where the hidden state of the GRU stores a running belief about the matching scores by matching features in the observations it has seen. One issue that arises is that we now have to define an ordering on the input images, whereas the output should be independent of the image ordering. We tackle this issue by randomly permuting the image sequences during training while constraining the output to be the same. During inference, we empirically observe that the final output has very little variance with respect to ordering of the input image sequence.

### 4.2.4 3D Grid Reasoning

Once the fused grid $\mathcal{G}^p$ is constructed, a classical multi-view stereo approach would directly evaluate the photo-consistency at the grid locations by comparing the appearance of the individual views and extract the surface at voxels where the images agree. We model this step with a 3D UNet that transforms the fused grid $\mathcal{G}^p$ into $\mathcal{G}^o$. The purpose of this network is to use shape cues present in $\mathcal{G}^p$ such as feature matches and silhouettes as well as build in shape priors like smoothness and symmetries and knowledge about object classes enabling it to produce complete shapes even when only partial information is visible. The UNet architecture yet again allows the system to use large enough receptive fields for doing multi-scale matching while also using lower level information directly when needed to produce its final estimate $\mathcal{G}^o$. In the case of full 3D supervision (Voxel LSM), this grid can be made to represent

a per voxel occupancy map. $\mathcal{G}^o$ can also be seen as a feature grid containing the final representation of the 3D world our system produces from which views can be rendered using the projection operation described below.

### 4.2.5 Differentiable Projection

Given a 3D feature grid $G$ and a camera $\mathcal{P}$, the projection operation produces a 2D feature map $\mathcal{O}$ by gathering information along viewing rays. The direct method would be to trace rays for every pixel and accumulate information from all the voxels on the ray's path. Such an implementation would require handling the fact that different rays can pass through different number of voxels on their way. For example, one can define a reduction function along the rays to aggregate information (*e.g.* max, mean) but this would fail to capture spatial relationships between the ray features. Instead, we choose to adopt a plane sweeping approach where we sample from locations on depth planes at equally spaced z-values $\{z_k\}_{k=1}^{N_z}$ along the ray.

Consider a 3D point $X_w$ that lies along the ray corresponding to a 2D point $p$ in the projected feature grid at depth $z_w$ - *i.e.* $p = K[R|t]X_w$ and $z(X_w) = z_w$. The corresponding feature $\mathcal{O}(p)$ is computed by sampling from the grid $\mathcal{G}$ at the (continuous) location $X_w$. This sampling can be done differentiably in 3D using trilinear interpolation. In practice, we use nearest neighbor interpolation in 3D for computational efficiency. Samples along each ray are concatenated in ascending z-order to produce the 2D map $\mathcal{O}$ where the features are stacked along the channel dimension. Rays in this feature grid can be trivially traversed by just following columns along the channel dimension allowing us to *learn* the function to pool along these rays by using 1x1 convolutions on these feature maps and progressively reducing the number of feature channels.

### 4.2.6 Architecture Details

As mentioned above, we present two versions of LSMs - Voxel LSM (V-LSM) and Depth LSM (D-LSM). Given one or more images and cameras, Voxel LSM (V-LSM) produces a voxel occupancy grid whereas D-LSM produces a depth map per input view. Both systems share the same set of CNN architectures (UNet) for the image encoder, grid reasoning and the recurrent pooling steps. We use instance normalization for all our convolution operations and layer normalization for the 3D convolutional GRU. In V-LSM, the final grid $\mathcal{G}^o$ is transformed into a probabilistic voxel occupancy map $\mathcal{V} \in R^{v_h \times v_w \times v_d}$ by a 3D convolution followed by softmax operation. We use simple binary cross entropy loss between ground truth occupancy maps and $\mathcal{V}$. In D-LSM, $\mathcal{G}^o$ is first projected into 2D feature maps $\{\mathcal{O}_i\}_{i=1}^n$ which are then transformed into metric depth maps $\{d_i\}_{i=1}^n$ by 1x1 convolutions to learn the reduction function along rays followed by deconvolution layers to upsample the feature map back to the size of the input image. We use absolute $L_1$ error in depth to train D-LSM. We also

add skip connections between early layers of the image encoder and the last deconvolution layers producing depth maps giving it access to high frequency information in the images.

## 4.3 Experiments

In this section, we demonstrate the ability of LSMs to learn 3D shape reconstruction in a geometrically accurate manner. First, we present quantitative results for V-LSMs on the ShapeNet dataset [96] and compare it to various baselines, both classical and learning based. We then show that LSMs generalize to unseen object categories validating our hypothesis that LSMs go beyond object/class specific priors and use photo-consistency cues to perform category-agnostic reconstruction. Finally, we present qualitative and quantitative results from D-LSM and compare it to traditional multi-view stereo approaches.

### 4.3.1 Dataset and Metrics

We use the synthetic ShapeNet dataset [96] to generate posed image-sets, ground truth 3D occupancy maps and depth maps for all our experiments. More specifically, we use a subset of 13 major categories (same as [95]) containing around 44k 3D models resized to lie within the unit cube centered at the origin with a train/val/test split of $[0.7, 0.1, 0.2]$. We generated a large set of realistic renderings for the models sampled from a viewing sphere with $\theta_{az} \in [0, 360)$ and $\theta_{el} \in [-20, 30]$ degrees and random lighting variations. We also rendered the depth images corresponding to each rendered image. For the volumetric ground truth, we voxelize each of the models at a resolution of $32 \times 32 \times 32$. In order to evaluate the outputs of V-LSM, we binarize the probabilities at a fixed threshold (0.4 for all methods except visual hull (0.75)) and use the voxel intersection over union (IoU) as the similarity measure. To aggregate the per model IoU, we compute a per class average and take the mean as a per dataset measure. All our models are trained in a class agnostic manner.

**Implementation details.** We use $224 \times 224$ images to train LSMs with a shape batch size of 4 and 4 views per shape. Our world grid is at a resolution of $32^3$. We implemented our networks in Tensorflow and trained both the variants of LSMs for 100k iterations using Adam. The projection and unprojection operations are trivially implemented on the GPU with batched matrix multiplications and bilinear/nearest sampling enabling inference at around 30 models/sec on a GTX 1080Ti. We unroll the GRU for upto 4 time steps while training and apply the trained models for arbitrary number of views at test time.

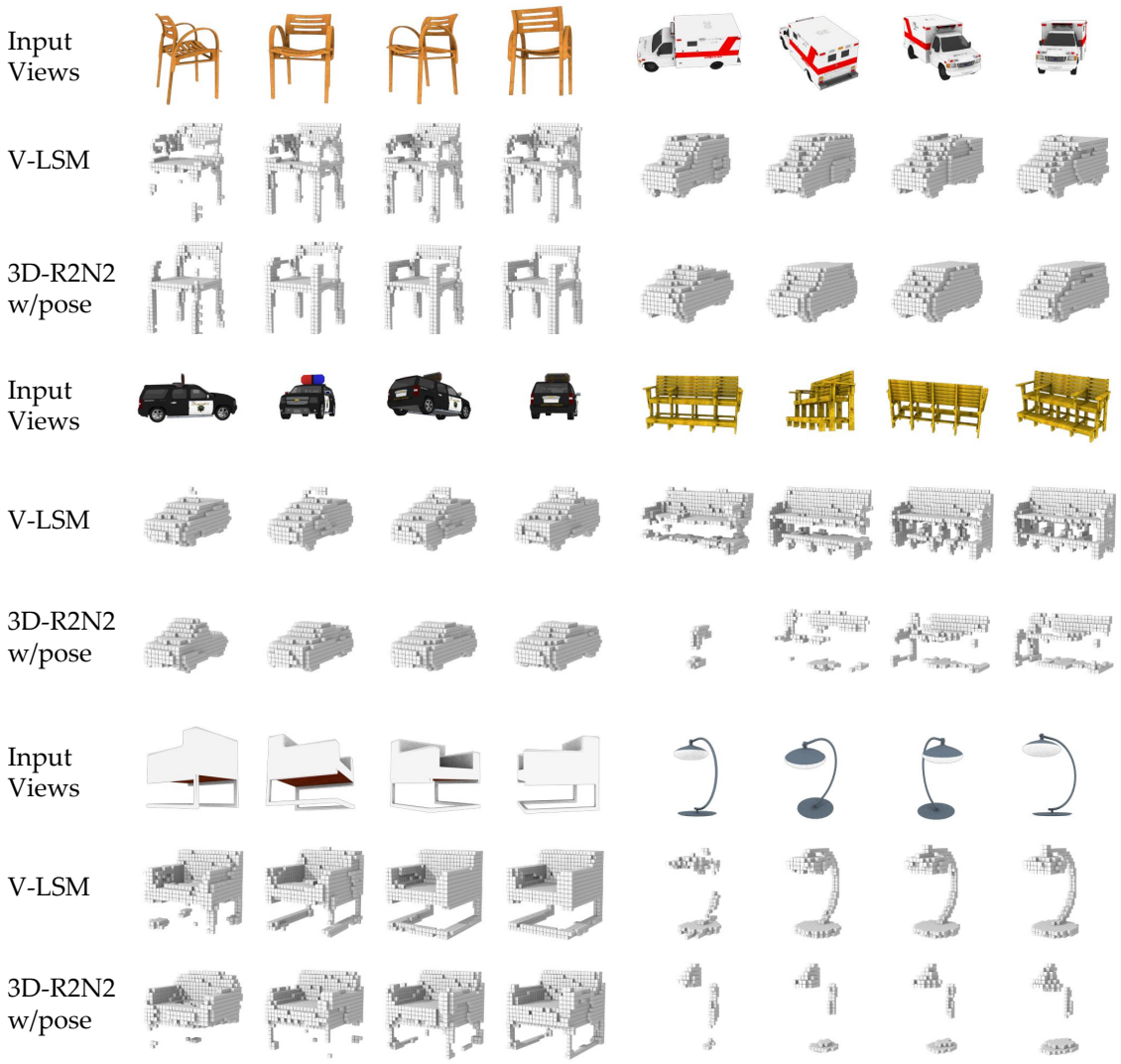


**Figure 4.3:** Voxel grids produced by V-LSM for example image sequences alongside a learning based baseline which uses pose information in a fully connected manner. V-LSM produces geometrically meaningful reconstructions (*e.g.* the curved arm rests instead of perpendicular ones (in R2N2) in the chair on the top left and the siren lights on top of the police car) instead of relying on purely semantic cues. More visualizations in supplementary material.

| # Views | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 3D-R2N2 [95] | 55.6 | 59.6 | 61.3 | 62.0 |
| Visual Hull | 18.0 | 36.9 | 47.0 | 52.4 |
| 3D-R2N2 w/pose | 55.1 | 59.4 | 61.2 | 62.1 |
| V-LSM | **61.5** | **72.1** | **76.2** | **78.2** |
| V-LSM w/bg | 60.5 | 69.8 | 73.7 | 75.6 |

**Table 4.1:** Mean Voxel IoU on the ShapeNet test set. Note that the original 3D-R2N2 system does not use camera pose whereas the 3D-R2N2 w/pose system is trained with pose information. V-LSM w/bg refers to voxel LSM trained and tested with random images as backgrounds instead of white backgrounds only.

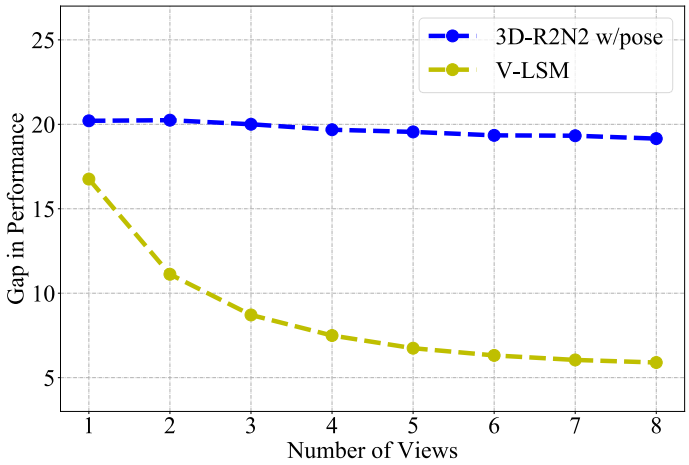


**Figure 4.4:** Generalization performance for V-LSM and 3D-R2N2 w/pose measured by gap in voxel IoU when tested on unseen object categories.

### 4.3.2 Multi-view Reconstruction on ShapeNet

We evaluate V-LSMs on the ShapeNet test set and compare it to the following baselines - a visual hull baseline which uses silhouettes to carve out volumes, 3D-R2N2 [95], a previously proposed system which doesn't use camera pose and performs multi-view reconstruction, 3D-R2N2 w/pose which is an extension of 3D-R2N2 where camera pose is injected using fully connected layers. For the experiments, we implemented the 3D-R2N2 system (and the 3D-R2N2 w/pose) and trained it on our generated data (images and voxel grids). Due to the difference in training data/splits and the implementation, the numbers are not directly comparable to the ones reported in [95] but we observe similar performance trends. For the 3D-R2N2 w/pose system, we use the camera pose quaternion as the pose representation and process


**Figure 4.5:** Qualitative results for per-view depth map prediction on ShapeNet. We show the depth maps predicted by Depth-LSM (visualized with shading from a shifted viewpoint) and the point cloud obtained by unprojecting them into world coordinates.

it through 2 fully connected layers before concatenating it with the feature passed into the LSTM. Table 4.1 reports the mean voxel IoU (across 13 categories) for sequences of $\{1, 2, 3, 4\}$ views. The accuracy increases with number of views for all methods but it can be seen that the jump is much less for the R2N2 methods indicating that it already produces a good enough estimate at the beginning but fails to effectively use multiple views to improve its reconstruction significantly. The R2N2 system with naively integrated pose fails to improve over the base version, completely ignoring it in favor of just image-based information. On the other hand, our system, designed specifically to exploit these geometric multi-view cues improves significantly with more views. Figure 4.3 shows some example reconstructions for V-LSM and 3D-R2N2 w/pose. Our system progressively improves based on the viewpoint it receives while the R2N2 w/pose system makes very confident predictions early on (sometimes "retrieving" a completely different instance) and then stops improving as much. As we use a geometric approach, we end up memorizing less and reconstruct when possible.

### 4.3.3 Generalization

In order to test how well LSMs learn to generalize to unseen data, we split our data into 2 parts with disjoint sets of classes - split 1 has data from 6 classes while split 2 has data from the other 7. We train three V-LSMs - trained on split 1 (V-LSM-S1), on split 2 (V-LSM-S2) and both splits combined (V-LSM-All). The quantity we are interested in is the change in performance when we test the system on a category it hasn't seen during training. We use the difference in test IoU of a category $C$ between V-LSM-All and V-LSM-S1 if $C$ is not in split 1 and vice versa. Figure 4.4 shows the mean of this quantity across all classes as the number of views change. It can be seen that for a single view, the difference in performance is fairly high and as we see more views, the difference in performance decreases - indicating that our system has learned to exploit category agnostic shape cues. On the other hand, the 3D-R2N2 w/pose system fails to generalize with more views. Note that the V-LSMs have been trained with a time horizon of 4 but are evaluated till upto 8 steps here.

### 4.3.4 Sensitivity to Noisy Camera Poses

We conducted experiments to quantify the effects of noisy camera pose and segmentations on performance for V-LSMs. We evaluated models trained with perfect poses on data with perturbed camera extrinsics and observed that performance degrades (as expected) yet still remains better than the baseline (at $10^\circ$ noise). We also trained new models with synthetically perturbed extrinsics and achieve significantly higher robustness to noisy poses while maintaining competitive performance (Figure 4.6). This is illustrated in Figure 4.6. The perturbation is introduced by generating a random rotation matrix which rotates the viewing axis by a max angular magnitude $\theta$ while still pointing at the object of interest.

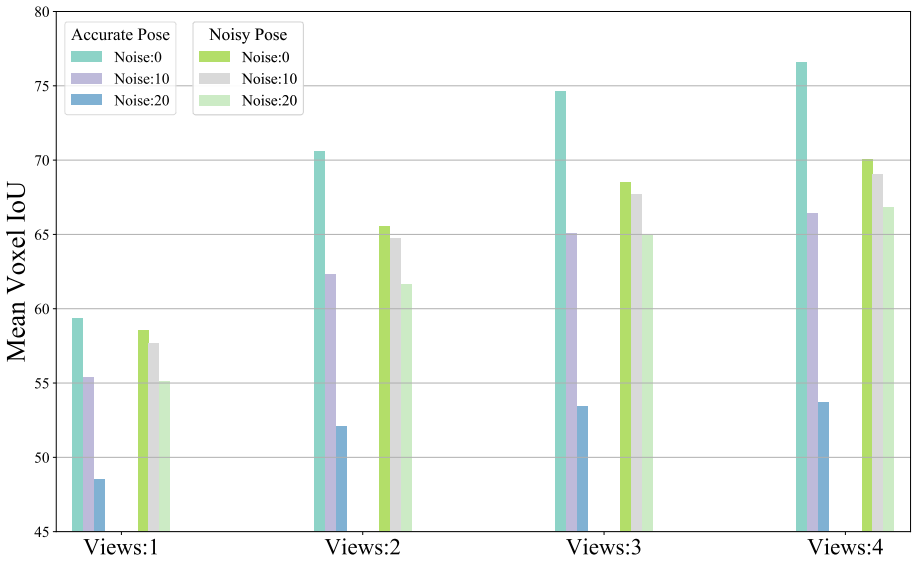


**Figure 4.6:** Sensitivity to noise in camera pose estimates for V-LSM for systems trained with and without pose perturbation.

We also trained LSMs on images with random images backgrounds (V-LSM w/bg in Table 4.1) rather than only white backgrounds and saw a very small drop in performance. This shows that our method learns to match features rather than relying heavily on perfect segmentations.

### 4.3.5 Multi-view Depth Map Prediction

We show qualitative results from Depth LSM in Figure 4.5. We manage to obtain thin structures in challenging examples (chairs/tables) while predicting consistent geometry for all the views. We note that the skip connections from the image to last layers for D-LSM do help in directly using low level image features while producing depth maps. The depth maps are viewed with shading in order to point out that we produce metrically accurate geometry. The unprojected point clouds also align well with each other showing the merits of jointly predicting the depth maps from a global volume rather than processing them independently.

### 4.3.6 Comparing D-LSM to Plane Sweeping

We qualitatively compare D-LSM to the popular plane sweeping (PS) approach [107, 108] for stereo matching. Figure 4.7 shows the unprojected point clouds from per view depths maps produced using PS and D-LSM using 5 and 10 images. We omit an evaluation with less images as plane sweeping completely fails with fewer images. We use the publicly available implementation for the PS algorithm [144] and

**(a)** PS 5 Views **(b)** LSM 5 Views **(c)** PS 10 Views **(d)** LSM 10 Views **(e)** PS 20 Views

**Figure 4.7:** Comparison between Depth-LSM and plane sweeping stereo (PS) with varying numbers of views.

use 5x5 zero mean normalized cross correlation as matching windows with 300 depth planes. We can see that our approach is able to produce much cleaner point clouds with less input images. It is robust to texture-less areas where traditional stereo algorithms fail (*e.g.* the car windows) by using shape priors to reason about them. We also conducted a quantitative comparison using PS and D-LSM with 10 views (D-LSM was trained using only four images). The evaluation region is limited to a depth range of $\pm\sqrt{3}/2$ (maximally possible depth range) around the origin as the original models lie in a unit cube centered at the origin. Furthermore, pixels where PS is not able to provide a depth estimate are not taken into account. Note that all these choices disadvantage our method. We compute the per depth map error as the median absolute depth difference for the valid pixels, aggregate to a per category mean error and report the average of the per category means for PS as 0.051 and D-LSM as 0.024.

### 4.3.7 Detailed Results on ShapeNet

We present per category voxel IoU numbers for V-LSMs (Table 4.2) and 3D-R2N2 w/pose (Table 4.3) for all 13 classes in ShapeNet. We also present per category results for the quantitative comparison between D-LSM and plane sweep stereo in Table 4.4.

| Classes | aero | bench | cabinet | car | chair | display | lamp | speaker | rifle | sofa | table | phone | vessel | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Views: 1** | 61.1 | 50.8 | 65.9 | 79.3 | 57.8 | 53.9 | 48.1 | 63.9 | 69.7 | 67.0 | 55.6 | 67.7 | 58.3 | 61.5 |
| **Views: 2** | 71.1 | 64.0 | 75.4 | 82.6 | 69.1 | 69.0 | 62.7 | 72.8 | 79.2 | 75.9 | 67.5 | 79.1 | 68.4 | 72.1 |
| **Views: 3** | 75.6 | 69.5 | 78.0 | 83.9 | 73.8 | 73.5 | 67.9 | 76.4 | 82.9 | 79.5 | 72.6 | 84.1 | 72.6 | 76.2 |
| **Views: 4** | 78.1 | 72.2 | 79.3 | 84.7 | 76.5 | 75.6 | 70.6 | 77.8 | 84.5 | 81.3 | 75.2 | 86.2 | 74.1 | 78.2 |

**Table 4.2:** Mean Voxel IoU for V-LSM for all classes in the ShapeNet test set.

| Classes | aero | bench | cabinet | car | chair | display | lamp | speaker | rifle | sofa | table | phone | vessel | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Views: 1** | 56.7 | 43.2 | 61.8 | 77.6 | 50.9 | 44.0 | 40.0 | 56.7 | 56.5 | 58.9 | 51.6 | 65.6 | 53.1 | 55.1 |
| **Views: 2** | 59.9 | 49.7 | 67.0 | 79.5 | 55.0 | 49.8 | 43.1 | 61.6 | 59.9 | 63.9 | 56.0 | 70.4 | 57 | 59.4 |
| **Views: 3** | 61.3 | 51.9 | 69.0 | 80.2 | 56.8 | 53.3 | 44.2 | 62.9 | 61.0 | 65.3 | 58.0 | 73.4 | 58.9 | 61.2 |
| **Views: 4** | 62.0 | 53.0 | 69.7 | 80.6 | 57.7 | 55.1 | 44.5 | 63.5 | 61.6 | 66.3 | 58.8 | 74.3 | 59.5 | 62.1 |

**Table 4.3:** Mean Voxel IoU for 3D-R2N2 w/pose for all classes in the ShapeNet test set.

| Classes | aero | bench | cabinet | car | chair | display | lamp | speaker | rifle | sofa | table | phone | vessel | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Plane Sweep** | 0.029 | 0.047 | 0.074 | 0.043 | 0.054 | 0.079 | 0.043 | 0.068 | 0.023 | 0.066 | 0.054 | 0.041 | 0.043 | 0.051 |
| **Depth LSM** | 0.024 | 0.030 | 0.022 | 0.016 | 0.023 | 0.028 | 0.027 | 0.029 | 0.023 | 0.021 | 0.025 | 0.021 | 0.027 | 0.024 |

**Table 4.4:** Mean depth map error ($L_1$ distance between predictions and ground truth at valid pixels) in the ShapeNet test set. Please refer to the main text for more details.

## 4.4 Discussion

We have presented Learnt Stereo Machines (LSM) - an end-to-end learnt system that performs multi-view stereopsis. The key insight of our system is to use ideas from projective geometry to differentiably transfer features between 2D images and the 3D world and vice-versa. In our experiments we showed the benefits of our formulation over direct methods - we are able to generalize to new object categories and produce compelling reconstructions with fewer images than classical systems. However, our system also has some limitations. We discuss some below and describe how they lead to future work.

A limiting factor in our current system is the coarse resolution ($32^3$) of the world grid. Classical algorithms typically work on much higher resolutions frequently employing special data structures such as octrees. We can borrow ideas from recent works [145, 146] which show that CNNs can predict such high resolution volumes. LSMs can, in principle, be applied to more general geometry than objects, *e.g.* multiple objects and entire scenes. The main challenge in this setup is to find the right global grid representation.

In our experiments we evaluated classical multi-view 3D reconstruction where the goal is to produce 3D geometry from images with known poses. However, our system is more general and the projection modules can be used wherever one needs to move between 2D image and 3D world frames. Instead of predicting just depth maps from our final world representation, one can also predict other view specific representations such as silhouettes or pixel wise part segmentation labels *etc.* We can also project the final world representation into views that we haven't observed as inputs. This can be used to perform (for example) view synthesis grounded in 3D.

# Chapter 5

# Conclusion

In this thesis, we have presented a number of advances towards learning based 3D object reconstruction. In Chapter 2, we presented an algorithm to learn category-specific deformable 3D models for diverse object categories and showed how to use them in conjunction with recognition systems to achieve fully automatic object localization and reconstruction from a single image. In Chapter 3, we worked towards calibrating the size of the reconstructed objects by inferring their relative sizes and depths from a single image. We did this by leveraging amodal bounding boxes and reasoning about object co-occurences in image collections. We worked towards unifying single and multi-view reconstruction with a CNN in Chapter 4 with Learnt Stereo Machines.

We have still just scratched the tip of the iceberg as alluded to in the limitations of current works in the chapters above. A number of challenges remain in tightly integrating semantic reasoning and 3D reconstruction (some of which we summarize below) and present for exciting directions for future work.

**Shape Representations:** In our works, we have explored using deformable meshes, voxel occupancy grids and depth maps as representations for shape. While each presents its own benefits, it is still unclear whether one has all the desirable properties for shape representations. For example, part compositionality plays a crucial role in the human visual system which none of the above representations exhibit. There have been promising works towards modelling shapes with primitives such as planes and simple shapes [147] (cubes, cylinders *etc.*). Automatic discovery of such primitives which can be shared across object categories would allow for greater interpretability in reconstructions.

**Learning without explicit supervision:** We, as humans, almost never have "ground truth" data to learn from - especially for 3D shapes. Our mental models are built from observing objects from different viewpoints, in various lighting conditions, by interacting with them *etc.* This remains a critical problem to solve for current

learning systems to scale beyond available 3D datasets. Some recent works [137, 139] have investigated using motion and projection into novel views as a proxy for learning shapes. An exciting direction to investigate would be coupling active exploration with 3D shape inference. For example, a robot could derive shape cues for an object by trying to grasp/poke/manipulate it in certain ways.

**Quality of reconstructions:** A common issue with current methods for learning-based shape inference methods is that they dont produce high quality outputs (in terms of accuracy, fidelity and resolution). This is in contrast with classical systems which produce extremely detailed models, albeit from far more images. Recent attempts at modelling high resolution grids with CNNs [146] present a promising direction towards detailed reconstructions with learning-based cues. A related task is modelling large spaces with such systems which couples with the question of what shape representations are amenable to such problems.

**Related tasks:** While inferring 3D shape from 2D views is an end in itself, it can be especially useful for a number of complementary and upstream tasks. For example, estimating lighting and reflectance in scenes could benefit from learning-based 3D inference systems, particularly when learnt jointly [35]. More examples of properties for which are difficult to *ground-truth* and tightly coupled with 3D reasoning are optical flow and scene flow. Learning systems could provide a very natural way of jointly modelling these variety of tasks with strong inter-dependencies. Implicit 3D reasoning could also play a critical role in end-to-end learning of policies in robotics [148], *e.g.* for navigating in 3D scenes or manipulating objects.

# Bibliography


[1] Stephen E Palmer. *Vision science: Photons to phenomenology*. MIT press Cambridge, MA, 1999. 1

[2] H.G. Louis. Methods and apparatus for correlating corresponding points in two images, December 13 1960. US Patent 2,964,642. 1

[3] E. Kruppa. Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung. *Sitzungsberichte der Mathematisch Naturwissenschaftlichen Kaiserlichen Akademie der Wissenschaften*, 122:1939–1948, 1913. 1

[4] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981. 1

[5] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992. 1

[6] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 1, 36

[7] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 1

[8] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 1

[9] Avanish Kushal, Ben Self, Yasutaka Furukawa, David Gallup, Carlos Hernandez, Brian Curless, and Steven M Seitz. Photo tours. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 57–64. IEEE, 2012. 1

[10] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Computer Graphics and Interactive Techniques*, 1999. 1, 5, 38

[11] Ira Kemelmacher-Shlizerman. Internet based morphable model. In *International Conference on Computer Vision*, 2011. 1

[12] Supasorn Suwajanakorn, Ira Kemelmacher-Shlizerman, and StevenM. Seitz. Total moving face reconstruction. In *European Conference on Computer Vision*. 2014. 1

[13] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[14] Yang LeCun, B. Boser, J.S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to hand-written zip code recognition. In *Neural Computation*, 1989. 1, 25

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2, 6, 7, 24, 26

[16] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2, 6, 7

[17] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2015. 2, 7, 14, 19

[18] Shubham Tulsiani, Abhishek Kar, João Carreira, and Jitendra Malik. Learning category-specific deformable 3d models for object reconstruction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. 4, 38

[19] Chetan Nandakumar, Antonio Torralba, and Jitendra Malik. How little do we need for 3-d shape perception? *Perception-London*, 2011. 4

[20] Ramakant Nevatia and Thomas O Binford. Description and recognition of curved objects. *Artificial Intelligence*, 1977. 4

[21] Abhinav Gupta, Alexei A Efros, and Martial Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *European Conference on Computer Vision*. 2010. 4, 22

[22] Jianxiong Xiao, Bryan Russell, and Antonio Torralba. Localizing 3d cuboids in single-view images. In *Advances in Neural Information Processing Systems*, 2012. 4

[23] Lawrence Gilman Roberts. *Machine Perception of Three-Dimensional Solids*. PhD thesis, Massachusetts Institute of Technology, 1963. 4

[24] Joseph J Lim, Hamed Pirsiavash, and Antonio Torralba. Parsing ikea objects: Fine pose estimation. In *International Conference on Computer Vision*, 2013. 4

[25] Scott Satkin, Maheen Rashid, Jason Lin, and Martial Hebert. 3dnn: 3d nearest neighbor. *International Journal of Computer Vision*, 2014. 4

[26] Bojan Pepik, Michael Stark, Peter Gehler, Tobias Ritschel, and Bernt Schiele. 3d object class detection in the wild. In *Workshop on 3D from a Single Image (3DSI) (in conjunction with CVPR'15)*, 2015. 4

[27] Hao Su, Qixing Huang, Niloy J Mitra, Yangyan Li, and Leonidas Guibas. Estimating image depth using shape collections. *ACM Transactions on Graphics (TOG)*, 33, 2014. 5

[28] Qixing Huang, Hai Wang, and Vladlen Koltun. Single-view reconstruction via joint analysis of image and shape collections. *ACM Transactions on Graphics (TOG)*, 34, 2015. 5

[29] Minhyuk Sung, Vladimir G Kim, Roland Angst, and Leonidas Guibas. Data-driven structural priors for shape completion. *ACM Transactions on Graphics (TOG)*, 2015. 5

[30] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: Shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, 2005. 5

[31] M Zeeshan Zia, Michael Stark, Bernt Schiele, and Konrad Schindler. Detailed 3d representations for object recognition and modeling. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2013. 5

[32] Derek Hoiem, Alexei A Efros, and Martial Hebert. Automatic photo pop-up. *ACM Transactions on Graphics (TOG)*, 2005. 5

[33] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *PAMI*, 2009. 5

[34] Kevin Karsch, Zicheng Liao, Jason Rock, Jonathan T. Barron, and Derek Hoiem. Boundary cues for 3d object shape recovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 5

[35] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2015. 5, 15, 17, 19, 53

[36] B.K.P. Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. PhD thesis, Massachusetts Inst. of Technology, 1970. 5

[37] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, 2014. 5, 38

[38] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *International Conference on Computer Vision*, 2015. 5

[39] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 5

[40] Sara Vicente, Joao Carreira, Lourdes Agapito, and Jorge Batista. Reconstructing pascal voc. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 5, 7, 8, 10, 15, 16, 17

[41] T.J. Cashman and A.W. Fitzgibbon. What shape are dolphins? building 3d morphable models from 2d images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2013. 5, 10

[42] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*. 2014. 6, 12, 19, 20

[43] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 2010. 6, 16, 25

[44] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2000. 7, 8

[45] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd. Coarse-to-fine low-rank structure-from-motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 8

[46] L. Torresani, A. Hertzmann, and C. Bregler. Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2008. 8

[47] Ravi Garg, Anastasios Roussos, and Lourdes Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 8

[48] Shengqi Zhu, Li Zhang, and Brandon Smith. Model evolution: An incremental approach to non-rigid structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 8

[49] Mukta Prasad, Andrew Fitzgibbon, Andrew Zisserman, and Luc Van Gool. Finding nemo: Deformable object class modelling using curve matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 8

[50] Mohsen Hejrati and Deva Ramanan. Analyzing 3d objects in cluttered images. In *Advances in Neural Information Processing Systems*, 2012. 8

[51] Sara Vicente and Lourdes de Agapito. Balloon shapes: Reconstructing and deforming objects with volume from images. In *3DV*, 2013. 8

[52] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1994. 10, 38

[53] Carlos Hernández Esteban and Francis Schmitt. Silhouette and stereo fusion for 3d object modeling. *CVIU*, 2004. 10, 12

[54] Yusuf Sahilliolu and Ycel Yemez. A surface deformation framework for 3d shape recovery. In *Multimedia Content Representation, Classification and Security*. 2006. 10

[55] Yu Chen, Tae-Kyun Kim, and Roberto Cipolla. Inferring 3d shapes and deformations from single views. In *European Conference on Computer Vision*, 2010. 10

[56] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2003. 10

[57] Jonathan T Barron and Jitendra Malik. Color constancy, intrinsic images, and shape estimation. *European Conference on Computer Vision*, 2012. 12, 15, 16, 17

[58] Nathaniel R Twarog, Marshall F Tappen, and Edward H Adelson. Playing with puffball: simple scale-invariant inflation for use in vision and graphics. In *ACM Symposium on Applied Perception*, 2012. 15, 16, 17, 19

[59] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting people using mutually consistent poselet activations. In *European Conference on Computer Vision*, 2010. 16

[60] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision*, 2011. 16

[61] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*, 2014. 16, 20, 25, 26

[62] Nicolas Aspert, Diego Santa-Cruz, and Touradj Ebrahimi. Mesh: Measuring errors between surfaces using the hausdorff distance. In *ICME*, 2002. 17

[63] Zhirong Wu, Shuran Song, Aditya Khosla, F Yu, L Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2015. 20

[64] Abhishek Kar, Shubham Tulsiani, João Carreira, and Jitendra Malik. Amodal completion and size constancy in natural scenes. In *International Conference on Computer Vision*. 2015. 21

[65] Harry Edwin Burton. The optics of euclid. *J. Opt. Soc. Am.*, 1945. 22

[66] Derek Hoiem, Alexei A Efros, and Martial Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 2008. 22, 28

[67] D. Hoiem and S. Savarese. *Representations and techniques for 3D object recognition and scene interpretation*. Morgan & Claypool Publishers, 2011. 22

[68] Jean-François Lalonde, Derek Hoiem, Alexei A Efros, Carsten Rother, John Winn, and Antonio Criminisi. Photo clip art. In *ACM Transactions on Graphics (TOG)*, 2007. 23

[69] Bryan C Russell and Antonio Torralba. Building a database of 3d scenes from user annotations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 23

[70] Gaetano Kanizsa. *Organization in vision: Essays on Gestalt perception*. Praeger Publishers, 1979. 23

[71] Thomas F Shipley and Philip J Kellman. *From fragments to objects: Segmentation and grouping in vision*, volume 130. Elsevier, 2001. 23

[72] Toby P Breckon and Robert B Fisher. Amodal volume completion: 3d visual completion. *Computer Vision and Image Understanding*, 2005. 23

[73] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2010. 24

[74] Golnaz Ghiasi, Yi Yang, Deva Ramanan, and Charless C Fowlkes. Parsing occluded people. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 24

[75] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Data-driven 3d voxel patterns for object category recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 24

[76] M Zeeshan Zia, Michael Stark, and Konrad Schindler. Towards scene understanding with detailed 3d object representations. *International Journal of Computer Vision*, 2014. 24

[77] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, pages 193–202, 1980. 25

[78] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012. 25, 32

[79] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 25

[80] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 27, 32

[81] William H Ittelson. Size as a cue to distance: Static localization. *The American Journal of Psychology*, 1951. 30

[82] JC Baird. Retinal and assumed size cues as determinants of size and distance perception. *Journal of Experimental Psychology*, 1963. 30

[83] William Epstein. The influence of assumed size on apparent distance. *The American Journal of Psychology*, 1963. 30

[84] Talia Konkle and Aude Oliva. Canonical visual size for real-world objects. *Journal of Experimental Psychology: human perception and performance*, 2011. 30

[85] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2000. 32

[86] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision algorithms: theory and practice*. Springer, 2000. 32

[87] Ling-Ling Wang and Wen-Hsiang Tsai. Camera calibration by vanishing lines for 3-d computer vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1991. 32

[88] Bruno Caprile and Vincent Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 1990. 32

[89] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, 2014. 32

[90] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in Neural Information Processing Systems*. 2017. 36

[91] T Kanade, H Kano, S Kimura, A Yoshida, and K Oda. Development of a video-rate stereo machine. In *International Conference on Intelligent Robots and Systems (IROS)*, 1995. 36

[92] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International Journal of Computer Vision (IJCV)*, 38(3):199–218, 2000. 36, 38

[93] Suzanna Becker and Geoffrey E Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 1992. 37

[94] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *International Conference on Computer Vision (ICCV)*, 2017. 37, 38, 39

[95] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision (ECCV)*, 2016. 37, 38, 41, 42, 44, 46

[96] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 37, 44

[97] David Marr and Tomaso Poggio. Cooperative computation of stereo disparity. In *From the Retina to the Neocortex*, pages 239–243. 1976. 37

[98] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision (IJCV)*, 47(1-3):7–42, 2002. 37

[99] George Vogiatzis, Philip HS Torr, and Roberto Cipolla. Multi-view stereo via volumetric graph-cuts. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 38

[100] Sudipta N Sinha, Philippos Mordohai, and Marc Pollefeys. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *International Conference on Computer Vision (ICCV)*, 2007. 38

[101] Daniel Cremers and Kalin Kolev. Multiview stereo and silhouette consistency via convex functionals over convex domains. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(6):1161–1174, 2011. 38

[102] Pau Gargallo, Emmanuel Prados, and Peter Sturm. Minimizing the reprojection error in surface reconstruction from images. In *International Conference on Computer Vision (ICCV)*, pages 1–8, 2007. 38

[103] Thomas Pollard and Joseph L Mundy. Change detection in a 3-d world. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 38

[104] Shubao Liu and David B Cooper. Statistical inverse ray tracing for image-based 3d modeling. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(10):2074–2088, 2014. 38

[105] Ali Osman Ulusoy, Andreas Geiger, and Michael J Black. Towards probabilistic volumetric reconstruction using ray potentials. In *International Conference on 3D Vision (3DV)*, 2015. 38

[106] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2005. 38

[107] Robert T Collins. A space-sweep approach to true multi-image matching. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 1996. 38, 49

[108] Ruigang Yang, Greg Welch, and Gary Bishop. Real-time consensus-based scene reconstruction using commodity graphics hardware. In *Computer Graphics Forum*, 2003. 38, 49

[109] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision (IJCV)*, 59(3):207–232, 2004. 38

[110] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2010. 38

[111] Marc Pollefeys, David Nistér, J-M Frahm, Amir Akbarzadeh, Philippos Mordohai, Brian Clipp, Chris Engels, David Gallup, S-J Kim, Paul Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision (IJCV)*, 78(2):143–167, 2008. 38

[112] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. In *International Conference on Computer Vision (ICCV)*, 2007. 38

[113] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Conference on Computer Graphics and Interactive Techniques*, 1996. 38

[114] Victor Lempitsky and Yuri Boykov. Global optimization for shape fitting. In *Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2007. 38

[115] Christopher Zach, Thomas Pock, and Horst Bischof. A globally optimal algorithm for robust tv-l 1 range image integration. In *International Conference on Computer Vision, (ICCV)*, 2007. 38

[116] Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In *International Conference on Computer Vision, (ICCV)*, 2007. 38

[117] Amaury Dame, Victor A Prisacariu, Carl Y Ren, and Ian Reid. Dense reconstruction using 3d object shape priors. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 38

[118] Sid Yingze Bao, Manmohan Chandraker, Yuanqing Lin, and Silvio Savarese. Dense object reconstruction with semantic priors. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 38

[119] Christian Häne, Nikolay Savinov, and Marc Pollefeys. Class specific 3d object shape priors using surface normals. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 38

[120] Abhishek Kar, Shubham Tulsiani, João Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2015. 38

[121] Christian Häne, Christopher Zach, Andrea Cohen, Roland Angst, and Marc Pollefeys. Joint 3d scene reconstruction and class segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 38

[122] Christian Haene, Christopher Zach, Andrea Cohen, and Marc Pollefeys. Dense semantic 3d reconstruction. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016. 38

[123] Nikolay Savinov, Christian Häne, Lubor Ladicky, and Marc Pollefeys. Semantic 3d reconstruction with continuous regularization and ray potentials using a visibility consistency constraint. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 38

[124] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research (JMLR)*, 17(1-32):2, 2016. 38

[125] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 38

[126] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Konrad Schindler, and Luc Van Gool. Learned multi-patch similarity. In *International Conference on Computer Vision, (ICCV)*, 2017. 38

[127] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 38

[128] Ashutosh Saxena, Jamie Schulte, and Andrew Y Ng. Depth estimation using monocular and stereo cues. In *Neural Information Processing Systems (NIPS)*, 2005. 38

[129] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 38

[130] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *European Conference on Computer Vision (ECCV)*, 2016. 38, 41

[131] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision (ECCV)*, 2016. 38

[132] Ayan Sinha, Jing Bai, and Karthik Ramani. Deep learning 3d shape surfaces using geometry images. In *European Conference on Computer Vision (ECCV)*, 2016. 38

[133] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 38

[134] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 39

[135] Ravi Garg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision (ECCV)*, 2016. 39

[136] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Neural Information Processing Systems (NIPS)*, 2016. 39

[137] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 39, 53

[138] Danilo Jimenez Rezende, SM Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *Neural Information Processing Systems (NIPS)*, 2016. 39

[139] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 39, 53

[140] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015. 41

[141] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Neural Information Processing Systems (NIPS)*, 2015. 41

[142] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997. 42

[143] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 2014. 42

[144] Christian Häne, Lionel Heng, Gim Hee Lee, Alexey Sizov, and Marc Pollefeys. Real-time direct dense matching on fisheye images using plane-sweeping stereo. In *International Conference on 3D Vision (3DV)*, 2014. 49

[145] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. Octnetfusion: Learning depth fusion from data. In *International Conference on 3D Vision (3DV)*, 2017. 51

[146] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *International Conference on 3D Vision (3DV)*, 2017. 51, 53

[147] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Computer Vision and Pattern Regognition (CVPR)*, 2017. 52

[148] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research (JMLR)*, 2016. 53