Learning 3D Object Categories by Looking Around Them

David Novotny[1,2] Diane Larlus[2] Andrea Vedaldi[1]
[1]Visual Geometry Group
Dept. of Engineering Science, University of Oxford
{david,vedaldi}@robots.ox.ac.uk
[2]Computer Vision Group
Naver Labs Europe
diane.larlus@naverlabs.com

Abstract

Traditional approaches for learning 3D object categories use either synthetic data or manual supervision. In this paper, we propose instead an unsupervised *method that is cued by observing objects from a moving vantage point. Our system builds on two innovations: a Siamese viewpoint factorization network that robustly aligns different videos together without explicitly comparing 3D shapes; and a 3D shape completion network that can extract the full shape of an object from partial observations. We also demonstrate the benefits of configuring networks to perform* probabilistic predictions *as well as of* geometry-aware data augmentation *schemes. We obtain state-of-the-art results on publicly-available benchmarks.*

1. Introduction

Despite their tremendous effectiveness in tasks such as object category detection, most deep neural networks do not understand the 3D nature of object categories. Reasoning about objects in 3D is necessary in many applications, for physical reasoning, or to understand the geometric relationships between different objects or scene elements.

The typical approach to learn 3D objects is to make use of large collections of high quality CAD models such as [5] or [42], which can be used to fully supervise models to recognize the objects' viewpoint and 3D shape. Alternatively, one can start from standard image datasets such as PASCAL VOC [8], augmented with other types of supervision, such as object segmentations and keypoint annotations [4]. Whether synthetically generated or manually collected, annotations have so far been required in order to overcome the significant challenges of learning 3D object categories, where both viewpoint and geometry are variable.

In this paper, we develop an alternative approach that can learn 3D object categories in an *unsupervised manner* (fig. 1), replacing synthetic or manual supervision with *motion*. Humans learn about the visual word by experiencing it continuously, through a variable viewpoint, which provides very strong cues on its 3D structure. Our goal is to build on such cues in order to learn the 3D geometry of object categories, using videos rather than images of objects. We are motivated by the fact that videos are almost as cheap as images to capture, and do not require annotations.

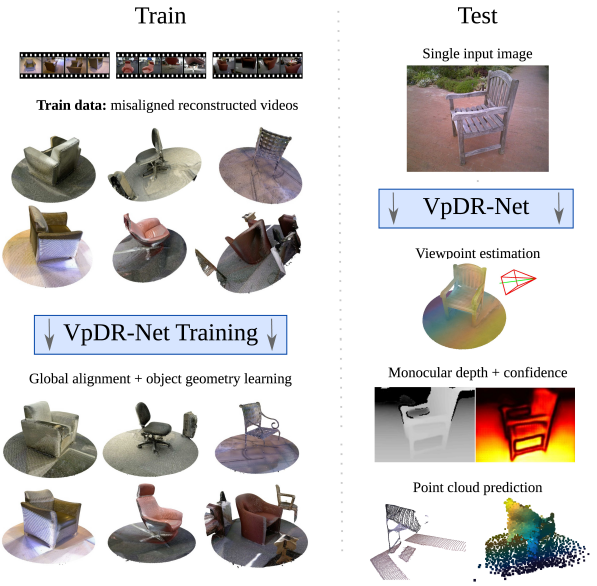

Figure 1. We propose a convolutional neural network architecture to learn the 3D geometry of object categories from videos only, without supervision. Once learned, the network can predict i) viewpoint, ii) depth, and iii) a point cloud, all from a single image of a new object instance.

We build on mature structure-from-motion (SFM) technology to extract 3D information from individual video sequences. However, these cues are specific to each object instance as contained in different videos. The challenge is to integrate this information in a global 3D model of the object category, as well as to work with noisy and incomplete reconstructions from SFM.

We propose a new deep architecture composed of three modules (fig. 2). The first module estimates the *absolute viewpoint* of objects in all video sequences (sec. 3.2). This aligns different object instances to a common reference frame where geometric relationships can be modeled more

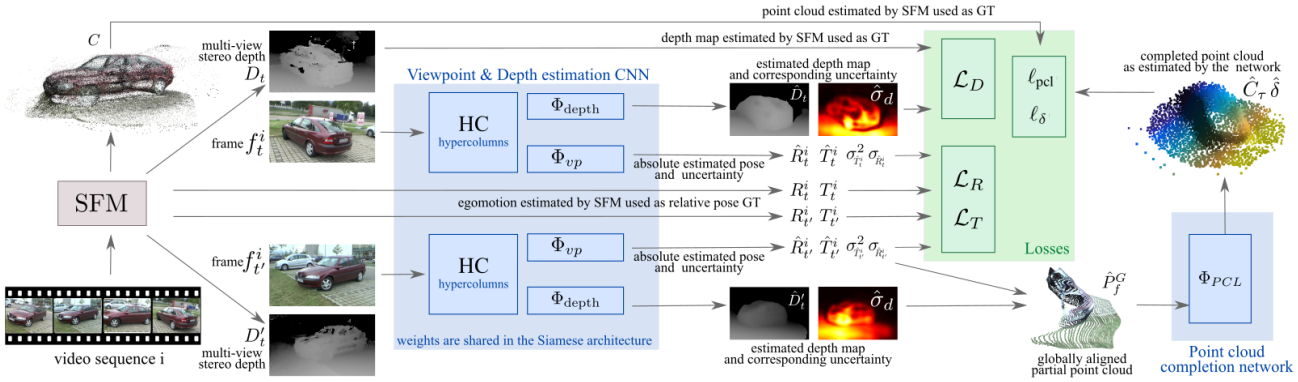

Figure 2. **Overview of our architecture.** As a preprocessing, structure from motion (SFM) extracts egomotion and a depth map for every frame. For training, our architecture takes pairs of frames f_t, $f_{t'}$ and produces a viewpoint estimate, a depth estimate, and a 3D geometry estimate. At test time, viewpoint, depth, and 3D geometry are predicted from single images.

easily. The second estimates the 3D shape of an object from a given viewpoint, producing a *depth map* (sec. 3.3). The third *completes the depth map to a full 3D reconstruction* in the globally-aligned reference frame (sec. 3.4). Combined and trained end-to-end without supervision, from videos alone, these components constitute VpDR-Net, a network for **v**iew**p**oint, **d**epth and **r**econstruction, capable of extracting viewpoint and shape of a new object instance from a single image.

One of our main contributions is thus to demonstrate the utility of using motion cues in learning 3D categories. We also introduce two significant technical innovations in the viewpoint and shape estimation modules as well as design guidelines and training strategies for 3D estimation tasks.

The first innovation (sec. 3.2) is a new approach to align video sequences of different 3D objects based on a *Siamese viewpoint factorization network*. While existing methods [37, 35] align shapes by looking at 3D features, we propose to train VpDR-Net to directly estimate the absolute viewpoint of an object. We train our network to reconstruct *relative camera motions* and we show that this implicitly aligns different objects instances together. By avoiding explicit shape comparisons in 3D space, this method is simpler and more robust than alternatives.

The second innovation (sec. 3.4) is a new network architecture that can generate a complete point cloud for the object from a partial reconstruction obtained from monocular depth estimation. This is based on a shape representation that predicts the support of a point probability distribution in 3D space, akin to a flexible voxelization, and a corresponding space occupancy map.

As a general design guideline, we demonstrate throughout the paper the utility of allowing deep networks to *express uncertainty* in their estimate by predicting probability distributions over outputs (sec. 3), yielding more robust training and useful cues (such as separating foreground and background in a depth map). We also demonstrate the significant power of *geometry-aware data augmentation*, where a deep network is used to predict the geometry of an image and the latter is used to generate new realistic views to train other components of the system (sec. 4). Each component and design choice is thoroughly evaluated in sec. 5, with significant improvements over the state-of-the-art.

2. Related work

Viewpoint estimation. The vast majority of methods for learning the viewpoint of object categories use manual supervision [32, 25, 11, 26, 43, 23, 39] or synthetic [36] data. In [40], a deep architecture predicts a relative camera pose and depth for a pair of images. Only a few works have used videos [37, 35]. [35] solves the shape alignment problem using a global search strategy based on the pairwise alignment of point clouds, a step we avoid by means of our Siamese viewpoint factorization network.

3D shape prediction. A traditional approach to 3D reconstruction is to use handcrafted 3D models [29, 21], and more recently 3D CAD models [5, 43]. Often the idea is to search for the 3D model in a CAD library that best fits the image [19, 1, 13, 2]. Alternatively, CAD models can be used to train a network to directly predict the 3D shape of an object [10, 41, 38, 7]. Morphable models have sometimes been used [45, 16], particularly for modeling faces [3, 20]. All these methods require 3D models at train time.

Data-driven approaches for geometry. Structure from motion (SFM) generally assumes fixed geometry between views and is difficult to apply directly to object categories due to intra-class variations. Starting from datasets of unordered images, methods such as [44] and [27] use SFM and manual annotations, such as keypoints in [4, 16], to estimate a rough 3D geometry of objects. Here, we leverage motion cues and do not need extra annotations.

3. Method

We propose a single Convolutional Neural Network (CNN), VpDR-Net, that learns a *3D object category* by observing it from a *variable viewpoint* in videos and no supervision (fig. 2). Videos do not solve the problem of modeling intra-class shape variations, but they provide powerful yet noisy cues about the 3D shape of individual objects.

VpDR-Net takes as an input a set of K video sequences $S^1, ..., S^K$ of an object category (such as cars or chairs), where a video $S^i = (f_1^i, ..., f_{N^i}^i)$ contains N^i RGB or RGBD frames $f_t^i \in \mathbb{R}^{H \times W \times \mathcal{C}}$ (where $\mathcal{C} = 3$ for RGB and $\mathcal{C} = 4$ for RGBD data) and learns a model of the 3D category. This model has three components: i) a predictor $\Phi_{\text{vp}}(f_i^t)$ of the *absolute viewpoint* of the object (implicitly aligning the different object instances to a common reference frame; sec. 3.2), ii) a *monocular depth* predictor $\Phi_{\text{depth}}(f_i^t)$ (sec. 3.3) and iii) and a *shape* predictor $\Phi_{\text{pcl}}(f_i^t)$ that extends the depth map to a point cloud capturing the complete shape of the object (sec. 3.4). Learning starts by preprocessing videos to extract instance-specific egomotion and shape information (sec. 3.1).

3.1. Sequence-specific structure and pose

Video sequences are pre-processed to extract from each frame f_t^i a tuple (K_t^i, g_t^i, D_t^i) consisting of: (i) the camera calibration parameters K_t^i, (ii) its pose $g_t^i \in SE(3)$, and (iii) a depth map $D_t^i \in \mathbb{R}^{H \times W}$ associating a depth value to each pixel of f_t^i. The camera pose $g_t^i = (R_t^i, T_t^i)$ consists of a rotation matrix $R_t^i \in SO(3)$ and a translation vector $T_t^i \in \mathbb{R}^3$.[1] We extract this information using off-the-shelf methods: the structure-from-motion (SFM) algorithm COLMAP for RGB sequences [33, 34], and an open-source implementation [31] of KinectFusion (KF) [24] for RGBD sequences. The information extracted from RGB or RGBD data is qualitatively similar, except that the scale of SFM reconstructions is arbitrary.

3.2. Intra-sequence alignment

Methods such as SFM or KF can reliably estimate camera pose and depth information for single objects and individual video sequences, but are not applicable to *different instances and sequences*. In fact, their underlying assumption is that geometry is fixed, which is true for single (rigid) objects, but false when the geometry and appearance differ due to intra-class variations.

Learning 3D object categories requires to relate their variable 3D shapes by identifying and putting in correspondence analogous geometric features, such as the object front and rear. For rigid objects, such correspondences can be expressed by rigid transformations that *align* occurrences of analogous geometric features.

The most common approach for aligning 3D shapes, also adopted by [35] for video sequences, is to extract and match 3D feature descriptors. Once objects in images or videos are aligned, the data can be used to supervise other tasks, such as learning a monocular predictor of the absolute viewpoint of an object [35].

One of our main contributions, described below, is to reverse this process by learning a viewpoint predictor *without* explicitly matching 3D shapes. Empirically (sec. 5), we show that, by skipping the intermediate 3D analysis, our method is often more effective and robust than alternatives.

Siamese network for viewpoint factorization. Geometric analogies between 3D shapes can often be detected in image space directly, based on visual similarity. Thus, we propose to train a CNN Φ_{vp} that maps a single frame f_t^i to its *absolute viewpoint* $\hat{g}_t^i = \Phi_{\text{vp}}(f_t^i)$ in the globally-aligned reference frame. We wish to learn this CNN from the viewpoints estimated by the algorithms of sec. 3.1 for each video sequence. However, these estimated viewpoints are *not* absolute, but valid only within each sequence; formally, there are unknown sequence-specific motions $h^i = (R^i, T^i) \in SE(3)$ that map the sequence-specific camera poses g_t^i to global poses $\hat{g}_t^i = g_t^i h^i$.[2]

To address this issue, we propose to supervise the network using *relative pose changes within each sequence*, which are invariant to the alignment transformation h^i. Formally, the transformation h^i is eliminated by computing the relative pose change of the camera from frame t to frame t':

$$\hat{g}_{t'}^i (\hat{g}_t^i)^{-1} = g_{t'}^i h^i (h^i)^{-1} (g_t^i)^{-1} = g_{t'}^i (g_t^i)^{-1}. \quad (1)$$

Expanding the expression with $\hat{g}_t^i = (\hat{R}_t^i, \hat{T}_t^i)$, we find equations expressing the relative rotation and translation

$$\hat{R}_{t'}^i (\hat{R}_t^i)^\top = R_{t'}^i (R_t^i)^\top, \quad (2)$$

$$\hat{T}_{t'}^i - R_{t'}^i (R_t^i)^\top \hat{T}_t^i = T_{t'}^i - R_{t'}^i (R_t^i)^\top T_t^i. \quad (3)$$

Eqs. (2) and (3) are used to constrain the training of a *Siamese architecture*, which, given two frames t and t', evaluates the CNN twice to obtain estimates $(\hat{R}_t^i, \hat{T}_t^i) = \Phi_{\text{vp}}(f_t^i)$ and $(\hat{R}_{t'}^i, \hat{T}_{t'}^i) = \Phi_{\text{vp}}(f_{t'}^i)$. The estimated poses are then compared to the ground truth ones, (R_t^i, T_t^i) and $(R_{t'}^i, T_{t'}^i)$, in a relative manner by using losses that enforce the estimated poses to satisfy eqs. (2) and (3):

$$\ell_R(\hat{R}_t^i, \hat{T}_t^i, \hat{R}_{t'}^i, \hat{T}_{t'}^i) \doteq \| \ln \hat{R}_{tt'}^i (R_{tt'}^i)^\top \|_F \quad (4)$$

$$\ell_T(\hat{R}_t^i, \hat{T}_t^i, \hat{R}_{t'}^i, \hat{T}_{t'}^i) \doteq \| \hat{T}_{tt'}^i - T_{tt'}^i \|_2 \quad (5)$$

where $\ln$ is the principal matrix logarithm and

$$R_{t't}^i \doteq R_{t'}^i (R_t^i)^\top, \qquad \hat{R}_{t't}^i \doteq \hat{R}_{t'}^i (\hat{R}_t^i)^\top,$$
$$T_{t't}^i \doteq T_{t'}^i - R_{t't}^i T_t^i, \qquad \hat{T}_{t't}^i \doteq \hat{T}_{t'}^i - \hat{R}_{t't}^i \hat{T}_t^i.$$

[1] We use the convention that g_t^i transforms world-relative coordinates p_{world} to camera-relative coordinates $p_{\text{camera}} = g_t^i p_{\text{world}}$.

[2] h^i composes to the right: it transforms the world reference frame and then moves it to the camera reference frame.

While this CNN is only required to correctly predict relative viewpoint changes *within each sequence*, since the *same CNN* is used for all videos, the most plausible/regular solution for the network is to assign similar viewpoint predictions $(\hat{R}^i_t, \hat{T}^i_t)$ to images viewed from the same viewpoint, leading to a globally consistent alignment of the input sequences. Furthermore, in a large family of 3D objects, different ones (e.g. SUVs and sedans) tend to be mediated by intermediate cases. This is shown empirically in sec. 5.

Scale ambiguity in SFM. For methods such as SFM, there is an additional ambiguity: reconstructions are known only up to sequence-specific scaling factors $\lambda^i > 0$, so that the camera pose is parametrized as $g^i_t(\lambda^i) = (R^i_t, \lambda^i T^i_t)$. This ambiguity leaves eq. (2) unchanged, but eq. (3) becomes:

$$\hat{T}^i_{t'} - \hat{R}^i_{t't}\hat{T}^i_t = \lambda^i(T^i_{t'} - R^i_{t't}T^i_t) \quad \Rightarrow \quad \hat{T}^i_{t't} = \lambda^i T^i_{t't}$$

During training, the ambiguity can be removed from loss (5) by dividing vectors $T^i_{t't}$ and $\hat{T}^i_{t't}$ by their Euclidean norm. Note that for KF sequences $\lambda^i = 1$. As the viewpoints are learned, an estimate of $\hat{\lambda}^i$ is computed using a moving average over training iterations for the other network modules to use (see supplementary material for details).

Probabilistic predictions. Due to intrinsic ambiguities in the images or to errors in the SFM supervision (caused for example by reflective or textureless surfaces), Φ_{vp} is occasionally unable to predict the ground truth viewpoint accurately. We found beneficial to allow the network to explicitly learn these cases and express uncertainty as an additional input-dependent prediction. For the translation component, we modify the network to predict the absolute pose $\hat{T}^i_t$ as well as its confidence score $\sigma_{\hat{T}^i_t}$ (predicted as the output of a soft ReLU units to ensure positivity). We then model the relative translation as a Gaussian distribution with standard deviation $\sigma_T = \sigma_{\hat{T}^i_{t'}} + \sigma_{\hat{T}^i_t}$ and our model is now learned by minimizing the negative log-likelihood $\mathcal{L}_T$ which replaces the loss ℓ_T:

$$\mathcal{L}_T = -\ln \frac{1}{(2\pi\sigma_T^2)^{\frac{3}{2}}} \exp\left(-\frac{1}{2}\frac{\ell_T^2}{\sigma_T^2}\right). \tag{6}$$

The rotation component is more complex due to the non-Euclidean geometry of $SO(3)$, but it was found sufficient to assume that the error term (4) has Laplace distribution and optimize $\mathcal{L}_R = -\ln \frac{1}{C_R} \exp\left(-\frac{\sqrt{2}\ell_R}{\sigma_R}\right)$, $\sigma_R = \sigma_{\hat{R}^i_{t'}} + \sigma_{\hat{R}^i_t}$, where C_R is a normalization term ensuring that the probability distribution integrates to one. During training, by optimizing the losses $\mathcal{L}_R$ and $\mathcal{L}_T$ instead of ℓ_R and ℓ_T, the network can discount gross errors by dividing the losses by a large predicted variance.

Architecture. The architecture of Φ_{vp} is a variant of ResNet-50 [15] with some modifications to improve its performance as viewpoint predictor. The lower layers of Φ_{vp} are used to extract a multiscale intermediate representation (denoted HC for hypercolumn [14] in fig. 2). The upper layers consist of 2×2 downsampling residual blocks that predict the viewpoint (see supp. material for details).

Figure 3. **Data augmentation**. Training samples generated leveraging monocular depth estimation (ours, top) and using depth from KF (baseline, bottom). Missing pixels due to missing depth in red.

3.3. Depth prediction

The depth predictor module Φ_{depth} of VpDR-Net takes individual frames f^i_t and outputs a corresponding depth map $\hat{D}_t = \Phi_{\text{depth}}(f^i_t)$, performing monocular depth estimation.

Estimating depth from a single image is inherently ambiguous and requires comparing the image to internal priors of the object shape. Similar to pose, we allow the network to explicitly *learn and express uncertainty* about depth estimates by predicting a posterior distribution over possible pixel depths. For robustness to outliers from COLMAP and KF, we assume a Laplace distribution with negative log-likelihood loss

$$\mathcal{L}_D = \sum_{j=1}^{WH} -\ln \frac{\sqrt{2}}{2\hat{\sigma}_{d_j}} \exp\left(-\frac{\sqrt{2}\,|d_j - \hat{\lambda^i}^{-1}\hat{d}_j|}{\hat{\sigma}_{d_j}}\right), \tag{7}$$

where d_j is the noisy ground truth depth output by SFM or KF for a given pixel j, and $\hat{d}_j$ and $\hat{\sigma}_{d_j}$ are respectively the corresponding predicted depth mean and standard deviation. The relative scale $\hat{\lambda}^i$ is 1 for KF and is estimated as explained in sec. 3.2 for SFM.

3.4. Point-cloud completion

Given any image f of an object instance, its *aligned 3D shape* can be reconstructed by estimating and aligning its depth map using the output of the viewpoint and depth predictors of sec. 3.2 and 3.3. However, since a depth map cannot represent the occluded portions of the object, such a reconstruction can only be partial. In this section we describe the third and last component of VpDR-Net, whose goal is to generate a full reconstruction of the object, beyond what is visible in the given view.

Partial point cloud. The first step is to convert the predicted depth map $\hat{D}_f = \Phi_{\text{depth}}(f)$ into a partial point cloud $\hat{P}_f \doteq \{\hat{p}_j : j = 1, \dots, HW\}$, $\hat{p}_j \doteq K^{-1}\begin{bmatrix} u_j & v_j & \hat{d}_i \end{bmatrix}^\top$, where (u_j, v_j) are the coordinates of a pixel j in the depth map $\hat{D}_f$ and K is the camera calibration matrix. Empirically, we have found that the reconstruction problem is

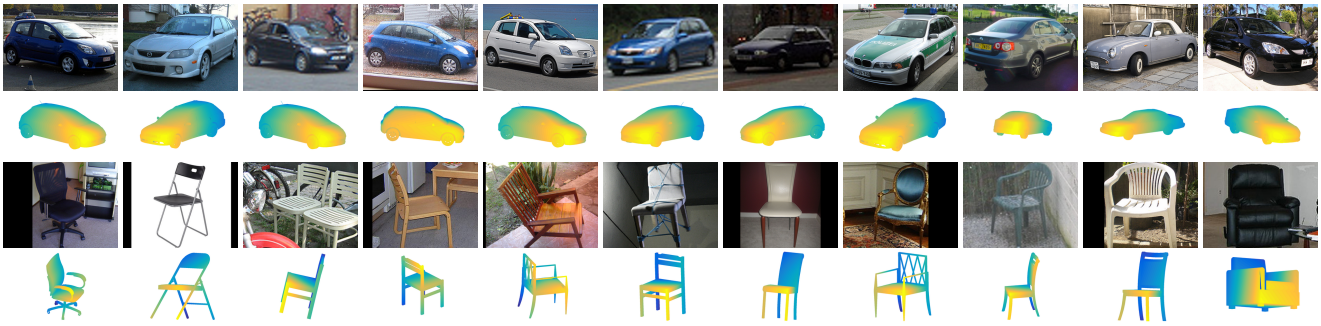

Figure 4. **Viewpoint prediction.** Most confident viewpoint predictions (sorted by predicted confidence from left to right) where the viewpoint predicted by VpDR-Net is used to align the Pascal3D ground truth CAD models with each image.

much easier if the data is aligned in the global reference frame established by VpDR-Net. Thus, we transform $\hat{P}_f$ into a globally-aligned point cloud as $\hat{P}_f^G = \hat{g}^{-1}\hat{P}_f$, where $\hat{g} = \Phi_{\text{vp}}(f)$ is the camera pose estimated by the viewpoint-prediction network.

Point cloud completion network. Next, our goal is to learn the point cloud completion part of our network Φ_{pcl} that takes the aligned but incomplete point could $\hat{P}_f^G$ and produces a complete object reconstruction $\hat{C}$. We do so by predicting a 3D occupancy probability field. However, rather than using a volumetric method that may require a discrete and fixed voxelization of space, we propose a simple and efficient alternative. First, the network Φ_{pcl} predicts a set of M 3D points $\hat{S} = (\hat{s}_1, \dots, \hat{s}_M) \in \mathbb{R}^{3\times M}$ that, during training, closely fit the ground truth 3D point cloud C. This step minimizes the fitting error:

$$\ell_{\text{pcl}}(\hat{S}) = \frac{1}{|C|}\sum_{c\in C}\min_{m=1,\dots,M}\|c - \hat{s}_m\|_2 \,. \tag{8}$$

The 3D point cloud $\hat{S}$ provides a good coverage of the ground truth object shape. However, this point cloud is conservative and distributed *in the vicinity* of the ground truth object. Thus, while this is not a precise representation of the object shape, it works well as a support of a probability distribution of space occupancy. In order to estimate the occupancy probability values, the network $\Phi_{\text{pcl}}(\hat{P}_f^G)$ predicts additional scalar outputs

$$\delta_m = |\{c \in C : \forall m' : \|\hat{s}_m - c\|_2 \le \|\hat{s}_{m'} - c\|_2\}|/|C|$$

proportional to the number of ground truth surface points $c \in C$ for which the support point $\hat{s}_m$ is the nearest neighbor. The network is trained to compute a prediction $\hat{\delta}_m$ of the occupancy masses δ_m by minimizing the squared error loss $\ell_\delta(\hat{\delta}, \delta) = \sum_{m=1}^M (\hat{\delta}_m - \delta_m)^2$.

Given the network prediction $(\hat{S}, \hat{\delta}) = \Phi_{\text{pcl}}(\hat{P}_f^G)$, the completed point cloud is then defined as the subset of points $\hat{C}$ that have sufficiently high occupancy, defined as: $\hat{C}_\tau = \{\hat{s}_m \in \hat{S} : \delta_m \ge \tau\}$ where τ is a confidence parameter. The set $\hat{C}_\tau$ can be further refined by using e.g. a 3D Laplacian filter to smooth out noise.

Architecture. The point cloud completion network Φ_{pcl} is modeled after PointNet [28], originally proposed to semantically *segment* a point clouds. Here we adapt it to perform a completely different task, namely 3D shape reconstruction. This is made possible by our model where shape is represented as a cloud of 3D support points $\hat{S}$ and their occupancy masses $\hat{\delta}$. Differently from Φ_{vp} and Φ_{depth}, Φ_{pcl} is *not* convolutional but uses a sequence of fully connected layers to process the 3D points in $\hat{P}_f^G$, after appending an appearance descriptor to each of them. A key step is to add an intermediate orderless pooling operator to remove the dependency on the order and number of input points (see the supplementary material for details). The architecture is configured to predict $M = 10^4$ points $\hat{S}$.

Leave out. During training the incomplete point cloud $\hat{P}_f^G$ is downsampled by randomly selecting between $M = 10^3$ and 10^4 points based on their depth prediction confidence as estimated by Φ_{depth}. Similar to dropout, dropping points allows the network to overfit less, to become less sensitive to the size of the input point cloud, and to implicitly discard background points (as these are assigned low confidence by depth prediction). For the latter reason, leave out is maintained at test time too with $M = 10^4$.

4. Geometry-aware data augmentation

As viewpoint prediction with deep networks benefits significantly from large training sets [36], we increase the effective size of the training videos by *data augmentation*. This is trivial for tasks such as classification, where one can translate or scale an image without changing its identity. The same is true for viewpoint recognition if the task is to only estimate the viewpoint orientation as in [36, 39], as images can be scaled and translated without changing the equivalent viewpoint orientation. However, this assumption is not satisfied if, as in our case, the goal is to estimate all 6 DoF of the camera pose.

Inspired by the approach of [12], we propose to solve

object class	test set	level of supervision	method	$\downarrow e_R$	$\downarrow e_C$	$\downarrow e_R^{rel}$	$\downarrow e_T^{rel}$	$\uparrow AP_{e_R}$	$\uparrow AP_{e_C}$
car	Pascal3D	unsupervised	VPNet + aligned FrC [35]	49.62	32.29	85.45	0.84	0.15	0.01
		unsupervised	**VpDR-Net + FrC (ours)**	**29.57**	**7.29**	**62.30**	**0.65**	**0.41**	**0.91**
		fully supervised	VPNet + Pascal3D	12.49	1.27	20.34	0.24	0.77	0.97
chair	Pascal3D	unsupervised	VPNet + aligned LDOS [35]	64.68	42.46	89.01	0.95	0.06	0.00
		unsupervised	**VpDR-Net + LDOS (ours)**	**42.34**	**16.72**	**71.35**	**0.93**	**0.23**	**0.22**
		fully supervised	VPNet + Pascal3D	34.37	6.14	67.41	0.74	0.26	0.66
	LDOS	unsupervised	VPNet + aligned LDOS [35]	**30.56**	0.61	71.40	0.77	0.30	0.18
		unsupervised	**VpDR-Net + LDOS (ours)**	33.92	**0.54**	**60.90**	**0.70**	**0.40**	**0.22**
		fully supervised	VPNet + Pascal3D	61.45	2.55	82.97	0.96	0.15	0.00

Table 1. **Viewpoint prediction.** Angular error e_r and camera-center distance e_c for absolute pose evaluation, and relative camera rotation error e_R^{rel} and translation error e_T^{rel} for relative pose evaluation. AP_{e_R} and AP_{e_C} evaluate absolute angular error and camera-center distance of the pose predictions taking into account the associated estimate confidence values. VpDR-Net trained on video sequences, is compared to VPNet trained on aligned video sequences and a fully-supervised VPNet. ↑ (resp. ↓) means larger (resp. lower) is better.

	$\downarrow e_R$	$\downarrow e_C$	$\downarrow e_R^{rel}$	$\downarrow e_T^{rel}$	$\uparrow AP_{e_R}$	$\uparrow AP_{e_C}$
Test set: **LDOS**						
VpDR-Net (ours)	**33.92**	**0.54**	**60.90**	**0.70**	**0.40**	**0.22**
VpDR-Net-NoProb	45.33	0.67	69.33	0.85	0.12	0.07
VpDR-Net-NoDepth	68.19	0.85	82.99	1.01	0.01	0.01
VpDR-Net-NoAug	35.16	0.59	63.54	0.73	0.38	0.19
Test set: **Pascal3D**						
VpDR-Net (ours)	**42.34**	**16.72**	**71.35**	0.93	**0.23**	**0.22**
VpDR-Net-NoProb	57.23	17.06	77.72	1.05	0.08	0.14
VpDR-Net-NoDepth	60.31	17.89	85.17	1.15	0.07	0.21
VpDR-Net-NoAug	43.52	18.80	72.93	**0.92**	0.10	0.17

Table 2. **Viewpoint prediction.** Different flavors of VpDR-Net with removed components to evaluate their respective impact.

this problem by using the estimated scene geometry to *generate new realistic viewpoints* (fig. 3). Given a sample (f_t^i, g_t^i, D_t^i), we apply a random perturbation to the viewpoint (with a forward bias to avoid unoccluding too many pixels) and use depth-image-based rendering (DIBR) [22] to generate a new sample (f_*^i, g_*^i, D_*^i), warping both the image and the depth map.

Sometimes the depth map D_t^i from KF contains too many holes to yield satisfactory DIBR results (fig. 3, bottom); we found preferable to use the depth $\hat{D}_t^i = \Phi_{\text{depth}}(f_t)$ estimated by the network which is less accurate but more robust, containing almost no missing pixels (fig. 3, top).

5. Experiments

We assess viewpoint estimation in sec. 5.1, depth prediction in sec. 5.2, and point cloud prediction in sec. 5.3.

Datasets. Throughout the experimental section, we consider three datasets for training and benchmarking our network: (1) **FreiburgCars (FrC)** [35] which consists of RGB video sequences with the camera circling around various types of cars; (2) the **Large Dataset of Object Scans (LDOS)** [6] containing RGBD sequences of man-made objects; and (3) **Pascal3D** [43], a standard benchmark for pose estimation [39, 36].

For viewpoint estimation, Pascal3D already contains viewpoint annotations. For LDOS, experiments focus on the *chair* class. In order to generate ground truth pose annotations for evaluation, we manually aligned 3D reconstructions of 10 randomly-selected chair videos and used 50 randomly-selected frames for each video as a test set.

For depth estimation, we evaluate on LDOS as it provides high quality depth maps one can use as ground truth.

For point cloud reconstruction, we use FrC and LDOS. Ground truth point clouds for evaluation are obtained by merging the SFM or RGBD depth maps from all frames of a given test video sequence, sampling $3{\cdot}10^4$ points and post-processing those using a 3D Laplacian filter. For FrC, five videos were randomly selected and removed from the train set, picking 60 random frames per video for evaluation. For LDOS the pose estimation test frames are used.

Learning details. VpDR-Net is trained with stochastic gradient descent with a momentum of 0.0005 and an initial learning rate of 10^{-2}. The weights of the losses were empirically set to achieve convergence on the training set. Better convergence was observed by training VpDR-Net in two stages. First, Φ_{depth} and Φ_{vp} were optimized jointly, lowering the learning rate tenfold when no further improvement in the training losses was observed. Then, Φ_{pcl} is optimized after initializing the bias of its last layer, which corresponds to an average point cloud of the object category, by randomly sampling points from the ground truth models.

5.1. Pose estimation

Pascal3D. First, we evaluate the VpDR-Net viewpoint predictor on the Pascal3D benchmark [43]. Unlike previous works [36, 39] that focus on estimating the object/camera viewpoint represented by a 3 DoF rotation matrix, we evaluate the full 6 DoF camera pose represented by the rotation matrix R together with the translation vector T.

In Pascal3D, the camera poses are expressed relatively to the whole scenes instead of the objects themselves, so we adjust the dataset annotations. We crop every object using bounding box annotations after reshaping the box to a fixed aspect ratio, and resize the crop to 240×320 pixels. The

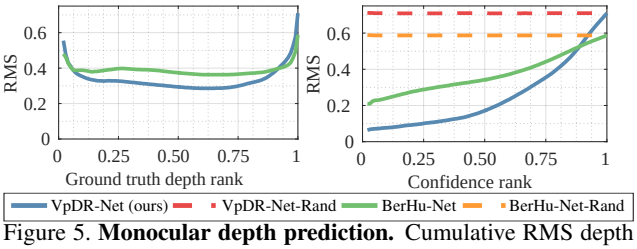

Figure 5. **Monocular depth prediction.** Cumulative RMS depth reconstruction error for the LDOS data, when pixels are ranked by ground truth depth (left) and by confidence (right).

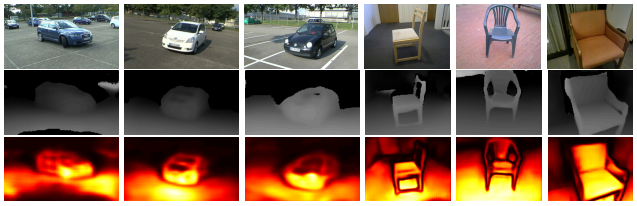

Figure 6. **Monocular depth prediction.** Top: input image; middle: predicted depth; bottom: predicted depth confidence. Depth maps are filtered by removing low confidence pixels.

camera pose is adjusted to the cropped object using the P3P algorithm to minimize the reprojection error between the camera-projected vertices of the ground truth CAD model and the original projection after cropping and resizing.

Absolute pose evaluation. We first evaluate absolute camera pose estimation using two standard measures: the angular error $e_R = 2^{-\frac{1}{2}} \| \ln R^* \hat{R}^\top \|_F$ between the ground truth camera pose R^* and the prediction $\hat{R}$ [39, 36], as well as the camera-center distance $e_C = \|\hat{C} - C^*\|_2$ between the predicted camera center $\hat{C}$ and the ground truth C^*. Following the common practice [39, 36] we report median e_R and e_C over all pose predictions on each test set.

Note that, while object viewpoints in Pascal3D and our method are internally consistent for a whole category, they may still differ between them by an arbitrary global 3D similarity transformation. Thus, as detailed in the supplementary material, the two sets of annotations are aligned by a single global similarity $\mathcal{T}_G$ before assessment.

Relative pose evaluation. To assess methods with measures independent of $\mathcal{T}_G$ we also evaluate: (1) the relative rotation error between pairs of ground truth relative camera motions $R^*_{tt'}$ and the corresponding predicted relative motions $\hat{R}_{tt'}$ given by $e^{\text{rel}}_R = 2^{-\frac{1}{2}} \| \ln R^*_{tt'} \hat{R}^\top_{tt'} \|_F$ and (2) the normalized relative translation error $e^{\text{rel}}_T = \|\hat{T}_{tt'} - T^*_{tt'}\|_2$, where both $\hat{T}_{tt'}$ and $T^*_{tt'}$ are ℓ_2-normalized so the measure is invariant to the scaling component of $\mathcal{T}_G$. We report the median errors over all possible image pairs in each test set.

Pose prediction confidence evaluation. A feature of our model is to produce confidence scores with its viewpoint estimates. We evaluate the reliability of these scores by correlating them with viewpoint prediction accuracy. In order to do so, predictions are divided into "accurate" and "inaccurate" by comparing their errors e_R and e_C to thresholds (set to $e_R = \frac{\pi}{6}$ following [36, 39] and $e_C = 15$ and 0.5 for Pascal3D or LDOS respectively). Predictions are then ranked by decreasing confidence scores and the average precisions AP_{e_R} and AP_{e_C} of the two ranked lists are computed.

Baselines. We compare our viewpoint predictor to a strong baseline, called **VPNet**, trained using absolute viewpoint labels. VPNet is a ResNet50 architecture [15] with the final softmax classifier replaced by a viewpoint estimation layer that predicts the 6 DoF pose $\hat{g}^i_t$. Following [39], rotation matrices are decomposed in Euler angles, each discretized in 24 equal bins. This network is trained to predict a softmax distribution over the angular bins and to regress a 3D vector corresponding to the camera translation T. The average softmax value across the three max-scoring Euler angles is used as a prediction confidence score.

We test both an unsupervised and a fully-supervised variant of VPNet. VPNet-unsupervised is comparable to our setting and is trained on the output of the global camera poses estimated from the videos by the state-of-the-art sequence-alignment method of [35]. In the fully-supervised setting, VPNet is trained instead by using ground-truth global camera poses provided by the Pascal3D training set.

Results. Table 1 compares VpDR-Net to the VPNet baselines. First, we observe that our baseline VPNet-unsupervised is very strong, as we report $e_R = 49.6$ error for the full rotation matrix, while the original method of [35] reports an error of 61.5 just for the azimuth component. Nevertheless, VpDR-Net outperforms VPNet in all performance metrics except for a single case (e_R for LDOS chairs). Furthermore, the advantage is generally substantial, and the unsupervised VpDR-Net reduces the gap with fully-supervised VPNet by 20 % or better in the vast majority of the cases. This shows the advantage of the proposed viewpoint factorization method compared to aligning 3D shapes as in [35]. Second, we observe that the confidence scores estimated by VpDR-Net are significantly more correlated with the accuracy of the predictions than the softmax scores in VPNet, providing a reliable self-assessment mechanism. The most confident viewpoint predictions of VpDR-Net are shown in fig. 4.

Ablation study. We evaluate the importance of the different components of VpDR-Net by turning them off and measuring performance on the *chair* class. In table 2, **VpDR-Net-NoProb** replaces the robust probabilistic losses $\mathcal{L}_R$ and $\mathcal{L}_T$ with their non-probabilistic counterparts ℓ_R and ℓ_T, and confidence predictions are replaced with random scores for AP evaluation. **VpDR-Net-NoDepth** removes the depth prediction and point cloud prediction branches during training, retaining only the Φ_{vp} subnetwork. **VpDR-Net-NoAug** does not use the data augmentation mechanism of sec. 4.

We observe a significant performance drop when each of the components is removed. This confirms the importance

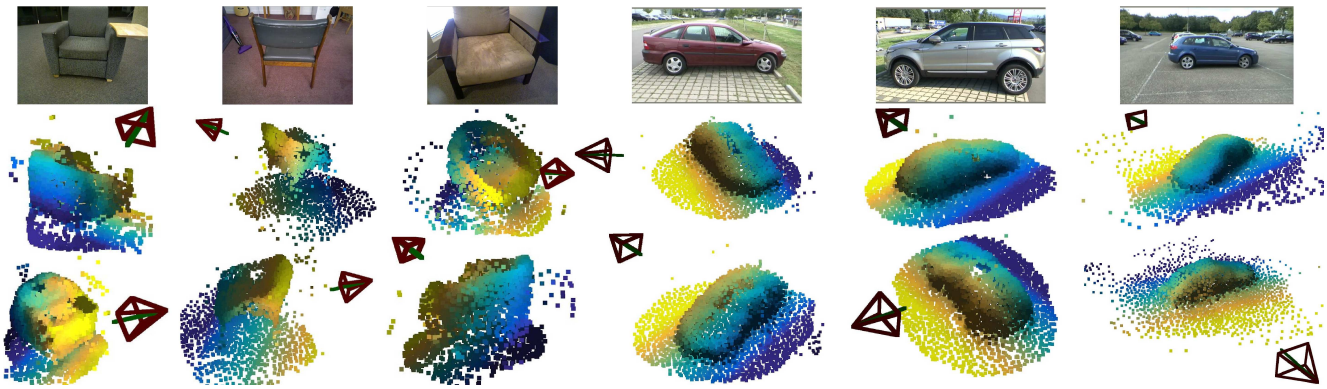

Figure 7. **Point cloud prediction.** From a single input image of an unseen object instance (top row), VpDR-Net predicts the 3D geometry of that instance in the form of a 3D point cloud (seen from two different angles, middle and bottom rows).

Test set	LDOS		FrC	
Metric	↑ mVIoU	↓ mD_{pcl}	↑ mVIoU	↓ mD_{pcl}
Aubry [1]	0.06	1.30	0.21	0.41
VpDR-Net (ours)	**0.13**	0.20	0.24	0.28
VpDR-Net-Fuse (ours)	**0.13**	**0.19**	**0.26**	**0.26**

Table 3. **Point cloud prediction**. Comparison between VpDR-Net and the method of Aubry *et al*. [1].

of all contributions in the network design. Interestingly, we observe that the depth prediction branch Φ_{depth} is crucial for pose estimation (*e.g*. -34.27 e_R on LDOS).

5.2. Depth prediction

The monocular depth prediction module of VpDR-Net is compared against three baselines: **VpDR-Net-Rand** uses VpDR-Net to estimate depth but predicts random confidence scores. **BerHu-Net** is a variant of the state-of-the-art depth prediction network from [18] based on the same Φ_{depth} subnetwork as VpDR-Net (but dropping Φ_{pcl} and Φ_{vp}). Following [18], for training it uses the BerHu depth loss and a dropout layer, which allows it to produce a confidence score of the depth measurements at test time using the sampling technique of [17, 9]. Finally, **BerHu-Net-Rand** is the same network, but predicting random confidence scores.

Results. Fig. 5 (right) shows the cumulative root-mean-squared (RMS) depth reconstruction error for LDOS after sorting pixels by their confidence as estimated by the network. By fitting better to inlier pixels and giving up on outliers, VpDR-Net produces a much better estimate than alternatives for the vast majority of pixels. Furthermore, accuracy is well predicted by the confidence scores. Fig. 5 (left) shows the cumulative RMS by depth, demonstrating that accuracy is better for pixels closer to the camera, which are more likely to be labeled with correct depth. Qualitative results are shown in fig. 6.

5.3. Point cloud prediction

We evaluate the point cloud completion module of VpDR-Net by comparing ground truth point clouds C to the point clouds $\hat{C}$ predicted by Φ_{pcl} using: (1) the voxel intersection-over-union (VIoU) measure that computes the Jaccard similarity between the volumetric representations of $\hat{C}$ and C, and (2) the normalized point cloud distance of [30]. We average these measures over the test set leading to mVIoU and mD_{pcl} (see supp. material for details).

VpDR-Net is compared against the approach of Aubry *et al*. [1] using their code. [1] is a 3D CAD model retrieval method which first trains a large number of exemplar models which, in our case, are represented by individual video frames with their corresponding ground truth 3D point clouds. Then, given a testing image, [1] detects the object instance and retrieves the best matching model from the database. We align the retrieved point cloud to the object location in the testing image using the P3P algorithm. For VpDR-Net, we evaluate two flavors. The original VpDR-Net that predicts the point cloud $\hat{C}$ and VpDR-Net-Fuse which further merges $\hat{C}$ with the predicted partial depth map point cloud $\hat{P}$.

Table 3 shows that our reconstructions are significantly better on both metrics for both LDOS chairs and FrC cars. Fusing the results with the original depth map produces a denser point cloud estimate and marginally improves the results. Qualitative results are shown in fig. 7.

6. Conclusion

We have demonstrated the power of motion cues in replacing manual annotations and synthetic data in learning 3D object categories. We have done so by proposing a single neural network that simultaneously performs monocular viewpoint estimation, depth estimation, and shape reconstruction. This network is based on two innovations, a new image-based viewpoint factorization method and a new probabilistic shape representation. The contribution of each component was assessed against suitable baselines.

Acknowledgments. We are grateful for support by NAVER LABS Europe and ERC 638009-IDIU.

References

[1] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proc. CVPR*, 2014. 2, 8

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

[3] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *PAMI*, 25(9):1063–1074, 2003. 2

[4] J. Carreira, S. Vicente, L. Agapito, and J. Batista. Lifting object detection datasets into 3d. *PAMI*, 38(7):1342–1355, 2016. 1, 2

[5] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. 1, 2

[6] S. Choi, Q. Zhou, S. Miller, and V. Koltun. A large dataset of object scans. *CoRR*, abs/1602.02481, 2016. 6

[7] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proc. ECCV*, 2016. 2

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 1

[9] Y. Gal and Z. Ghahramani. Bayesian convolutional neural networks with Bernoulli approximate variational inference. In *Proc. ICLR*, 2016. 8

[10] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *Proc. ECCV*, 2016. 2

[11] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *Proc. ICCV*, 2011. 2

[12] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proc. CVPR*, 2016. 5

[13] S. Gupta, P. A. Arbeláez, R. B. Girshick, and J. Malik. Aligning 3D models to RGB-D images of cluttered scenes. In *Proc. CVPR*, 2015. 2

[14] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proc. CVPR*, 2015. 4

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 4, 7

[16] A. Kar, S. Tulsiani, J. Carreira, and J. Malik. Category-specific object reconstruction from a single image. In *Proc. CVPR*, 2015. 2

[17] A. Kendall, V. Badrinarayanan, and R. Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *CoRR*, abs/1511.02680, 2015. 8

[18] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016. 8

[19] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing ikea objects: Fine pose estimation. In *Proc. ICCV*, 2013. 2

[20] F. Liu, D. Zeng, Q. Zhao, and X. Liu. Joint face alignment and 3d face reconstruction. In *Proc. ECCV*, 2016. 2

[21] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artif. Intell.*, 31(3):355–395, 1987. 2

[22] Y. Y. Morvan. *Acquisition, compression and rendering of depth and texture for multi-view video*. PhD thesis, Technische Universiteit Eindhoven, 2009. 6

[23] R. Mottaghi, Y. Xiang, and S. Savarese. A coarse-to-fine model for 3d pose estimation and sub-category recognition. In *Proc. CVPR*, 2015. 2

[24] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proc. ISMAR*, 2011. 3

[25] M. Ozuysal, V. Lepetit, and P.Fua. Pose estimation for category specific multiview object localization. In *Proc. CVPR*, 2009. 2

[26] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Multi-view priors for learning detectors from sparse viewpoint data. In *Proc. ICLR*, 2014. 2

[27] M. Prasad, A. Fitzgibbon, A. Zisserman, and L. V. Gool. Finding nemo: Deformable object class modelling using curve matching. In *Proc. CVPR*, 2010. 2

[28] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593, 2016. 5

[29] L. G. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology. Dept. of Electrical Engineering, 1963. 2

[30] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem. Completing 3d object shape from one depth image. In *Proc. CVPR*, 2015. 8

[31] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *Proc. ICRA*, 2011. 3

[32] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *Proc. ICCV*, 2007. 2

[33] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Proc. CVPR*, 2016. 3

[34] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proc. ECCV*, 2016. 3

[35] N. Sedaghat and T. Brox. Unsupervised generation of a viewpoint annotated car dataset from videos. In *Proc. ICCV*, 2015. 2, 3, 6, 7

[36] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proc. ICCV*, 2015. 2, 5, 6, 7

[37] i. Sun, H. Su, S. Savarese, and L. Fei-Fei. A multi-view probabilistic model for 3d object classes. In *Proc. CVPR*, 2009. 2

[38] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3d models from single images with a convolutional network. In *Proc. ECCV*, 2016. 2

[39] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *Proc. CVPR*, 2015. 2, 5, 6, 7

[40] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. *CoRR*, abs/1612.02401, 2016. 2

[41] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Proc. NIPS*, 2016. 2

[42] Y. Xiang, W. Kim, W. Chen, J. Ji, C. Choy, H. Su, R. Mottaghi, L. Guibas, and S. Savarese. Objectnet3d: A large scale database for 3d object recognition. In *Proc. ECCV*, 2016. 1

[43] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*, 2014. 2, 6

[44] S. Zhu, L. Zhang, and B. M. Smith. Model evolution: An incremental approach to non-rigid structure from motion. In *Proc. CVPR*, 2010. 2

[45] Z. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3d representations for object recognition and modeling. *PAMI*, 35(11):2608–2623, 2013. 2